NASA Conference Publication 10012
Part 2

NASA Workshop on Computational Structural Mechanics 1987

Nancy P. Sykes, *Editor*
Analytical Services and Materials, Inc.
Hampton, Virginia

Proceedings of a workshop sponsored by
the National Aeronautics and Space
Administration and held at
Langley Research Center
Hampton, Virginia
November 18–20, 1987

February 1989

(NASA-CP-10012-Pt-2) NASA WORKSHOP ON COMPUTATIONAL STRUCTURAL MECHANICS 1987, PART 2 (NASA. Langley Research Center) 374 p
CSCL 20K
N89-29789
--THRU--
N89-29798
Unclas
H1/39 0211957

National Aeronautics and
Space Administration

Preface

This document contains the proceedings of the NASA Workshop on Computational Structural Mechanics, held at NASA Langley Research Center, November 18-20, 1987. The workshop was sponsored jointly by NASA Langley Research Center and NASA Lewis Research Center.

The purpose of the workshop was to allow participants in Langley's and Lewis' Computational Structural Mechanics (CSM) research programs to meet and to share research objectives and accomplishments. The intent was to encourage a cooperative Langley/Lewis CSM program in which Lewis concentrates on engine structures applications, Langley concentrates on airframe and space structures applications, and all participants share technology of mutual interest.

The workshop was organized into the following three sessions:

I Concurrent Processing Methods and Applications

II Advanced Methods & Testbed/Simulator Development

III Computational Dynamics

Session I dealt with parallel processing methods and languages, new computer hardware, and software architecture to exploit parallel computers.

Session II dealt with the Langley CSM Testbed, the Lewis Engine Structures Computational Simulator, and Structural Analysis Technology involving finite elements, boundary elements, and probabilistic approaches.

Session III dealt with advanced methods for structural dynamics.

The use of trade names or names of manufacturers in this publication does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

W. Jefferson Stroud

With the exception of a few adjustments made primarily for the purpose of uniformity, all papers have been published as received.

—Editor

CONTENTS

PREFACE i
W. Jefferson Stroud, LaRC

Part 1*

OVERVIEW OF THE NASA PROGRAM
IN COMPUTATIONAL STRUCTURAL MECHANICS 1
Murray Hirschbein, NASA Headquarters

SESSION I - CONCURRENT PROCESSING METHODS AND APPLICATIONS

CSM PARALLEL STRUCTURAL METHODS RESEARCH 25
Olaf O. Storaasli, LaRC

COMPUTATIONAL STRUCTURAL METHODS AT NASA LEWIS 61
L. J. Kiraly, LeRC

TRANSPUTER FINITE ELEMENT SOLVER 75
Albert Danial, SPARTA

TRANSPUTER PARALLEL PROCESSING AT NASA LERC 107
Graham K. Ellis, ICOMP, LeRC

INNOVATIVE ARCHITECTURES
FOR DENSE MULTI-MICROPROCESSOR COMPUTERS 137
Robert E. Larson, Expert-EASE

PARALLEL LINEAR EQUATION SOLVERS
FOR FINITE ELEMENT COMPUTATIONS 171
James M. Ortega, Gene Poole, Courtenay Vaughan, Andrew Cleary,
Brett Averick, U. of Virginia

ALGORITHMS AND SOFTWARE FOR SOLVING FINITE ELEMENT EQUATIONS
ON SERIAL AND PARALLEL ARCHITECTURES 203
Alan George, U. of Tennessee

PARALLEL EIGENVALUE EXTRACTION 239
Fred A. Akl, Ohio U.

PARALLEL ALGORITHMS AND ARCHITECTURES FOR
COMPUTATIONAL STRUCTURAL MECHANICS 261
Merrell Patrick, Shing Ma, Umesh Mahajan, Duke U.

THE FORCE: A PORTABLE PARALLEL PROGRAMMING LANGUAGE
SUPPORTING COMPUTATIONAL STRUCTURAL MECHANICS 273
Harry F. Jordan, Muhammad S. Benten, Juergen Brehm,
Aruna Ramanan, U. of Colorado

* Published under separate cover

METHODS FOR DESIGN AND EVALUATION OF PARALLEL COMPUTING SYSTEMS (THE PISCES PROJECT) 281
Terrence W. Pratt, Robert Wise, Mary Jo Haught, U. of Virginia and ICASE, LaRC

MULTIPROCESSOR ARCHITECTURE: SYNTHESIS AND EVALUATION 299
Hilda M. Standley, U. of Toledo

ENVIRONMENTAL CONCEPT FOR ENGINEERING SOFTWARE ON MIMD COMPUTERS 323
L. A. Lopez and K. Valimohamed, U. of Illinois

HIERARCHIAL PARALLEL COMPUTER ARCHITECTURE DEFINED BY COMPUTATIONAL MULTIDISCIPLINARY MECHANICS 355
Joe Padovan, Doug Gute, Keith Johnson, U. of Akron

Part 2

SESSION II - ADVANCED METHODS & TESTBED/SIMULATOR DEVELOPMENT

S1
CSM RESEARCH: TESTBED DEVELOPMENT 387
Ronnie E. Gillian, LaRC

S2
CSM TESTBED ARCHITECTURE 419
Philip Underwood, Lockheed PARL

S3
COMPUTATIONAL STRUCTURAL MECHANICS ENGINE STRUCTURES COMPUTATIONAL SIMULATOR 459
C. C. Chamis, LaRC

S4
INTERFACING MODULES FOR INTEGRATING DISCIPLINE SPECIFIC STRUCTURAL MECHANICS CODES 487
Ned M. Endres, LaRC

S5
CSM RESEARCH: METHODS AND APPLICATION STUDIES 521
Norman F. Knight, Jr., LaRC

S6
GENERIC ELEMENT PROCESSOR (APPLICATION TO NONLINEAR ANALYSIS) 571
Gary Stanley, Lockheed PARL

S7
ASSESSMENT OF SPAR ELEMENTS AND FORMULATION OF SOME BASIC 2-D AND 3-D ELEMENTS FOR USE WITH TESTBED GENERIC ELEMENT PROCESSOR 653
Mohammad A. Aminpour, AS&M, LaRC

S8
DEVELOPMENT AND VERIFICATION OF LOCAL/GLOBAL ANALYSIS TECHNIQUES FOR LAMINATED COMPOSITES 683
O. Hayden Griffin, Jr., VPI&SU

S9
CONTROL OF THE ERRORS OF DISCRETIZATION AND IDEALIZATION IN FINITE ELEMENT ANALYSIS 733
Barna A. Szabó, Washington U.

Part 3*

BOUNDARY ELEMENTS FOR STRUCTURAL ANALYSIS 763
Ray Wilson, P&W

DEVELOPMENT OF AN INTEGRATED BEM FOR HOT FLUID-STRUCTURE INTERACTION 831
P. K. Banerjee and G. F. Dargush, SUNY-Buffalo

PROBABILISTIC STRUCTURAL ANALYSIS METHODS FOR SELECT SPACE PROPULSION SYSTEM STRUCTURAL COMPONENTS 865
T. A. Cruse, SWRI

PROBABILISTIC FINITE ELEMENTS (PFEM) APPLIED TO STRUCTURAL DYNAMICS AND FRACTURE MECHANICS 903
Wing-Kam Liu, Ted Belytschko, A. Mani, G. Besterfield, Northwestern U.

3-D INELASTIC ANALYSES FOR COMPUTATIONAL STRUCTURAL MECHANICS 943
D. A. Hopkins and C. C. Chamis, LeRC

SPECIALTY FUNCTIONS FOR SINGULARITY MECHANICS PROBLEMS 981
Nesrin Sarigul, Ohio State U.

SESSION III - COMPUTATIONAL DYNAMICS

LARC COMPUTATIONAL STRUCTURAL DYNAMICS OVERVIEW 1013
J. M. Housner, LaRC

ALGORITHMS AND SOFTWARE FOR NONLINEAR STRUCTURAL DYNAMICS 1043
Ted Belytschko, Noreen D. Gilbertsen, Mark O. Neal, Northwestern U.

CONCURRENT ALGORITHMS FOR TRANSIENT FE ANALYSIS 1067
M. Ortiz, Brown U. and B. Nour-Omid, Lockheed PARL

COMPUTATIONAL METHODS AND SOFTWARE SYSTEMS FOR DYNAMICS AND CONTROL OF LARGE SPACE STRUCTURES 1105
K. C. Park, C. A. Felippa, Charbel Farhat, J. D. Downer, J. C. Chiou, W. K. Belvin, U. of Colorado

MULTI-GRID FOR STRUCTURES ANALYSIS 1133
Albert F. Kascak, LeRC

* Published under separate cover

N89
29790
UNCLAS

N89-29790

CSM RESEARCH: TESTBED DEVELOPMENT

Ronnie E. Gillian†

NASA Langley Research Center

INTRODUCTION

The Computational Structural Mechanics(CSM) Activity at Langley Research Center is developing methods for structural analysis on modern computers. To facilitate that research effort, a Testbed Development environment is being constructed. It is the purpose of the Testbed to insulate researchers from differences in the computer operating systems of modern computer systems and permit concentrated effort on the analytical problem rather than the analytical tools required to solve that problem.

While modern computers enable the solution of larger problems of increasing complexity, they do so at a cost. Distributed computer environments, vector processing hardware and multiple processors dominate the current computer environment and threaten to overwhelm future analysis software. The systems software for current computers becomes more complex in order to manage the increasing complexity. The applications developer is caught between conflicting goals. They must take advantage of the computing power of new computer systems while maintaining a stable software development system. The CSM Testbed is being developed to address this problem for the computational structural analysis research community.

The Langley CSM activity was initiated in October 1984. This paper describes the current directions for the Testbed Development Team of the Langley CSM activity.

† Mathematician, Structural Mechanics Branch, Structures and Dynamics Division.

LANGLEY CSM PROGRAM

Parallel Structural Methods Research - Olaf O. Storaasli

Testbed Development - Ronnie Gillian

Methods and Applications Studies - Norm Knight

LANGLEY CSM PROGRAM

The CSM activity at Langley is divided into three activities, parallel structural methods, Testbed development, and methods research. This talk will describe the current activity in the Testbed Development area.

The Testbed Development area is located between the advanced computation area dominated by a computer science team, and the methods research dominated by structural engineers. It is an attempt to bridge the gap between these two disciplines and bring both groups closer to the realities imposed by implementation.

CSM TESTBED DEFINITION

A computer program for developing and evaluating advanced analysis methods and software architecture for a new generation of computers.

CSM TESTBED DEFINITION

The CSM Testbed is a computer program for developing and evaluating advanced analysis methods and software architecture for a new generation of computers. This definition emphasizes several different areas.

First, the Testbed is software. It is a computer program consisting of over 100,000 lines of FORTRAN source code with low level I/O in either "C" or assembly language.

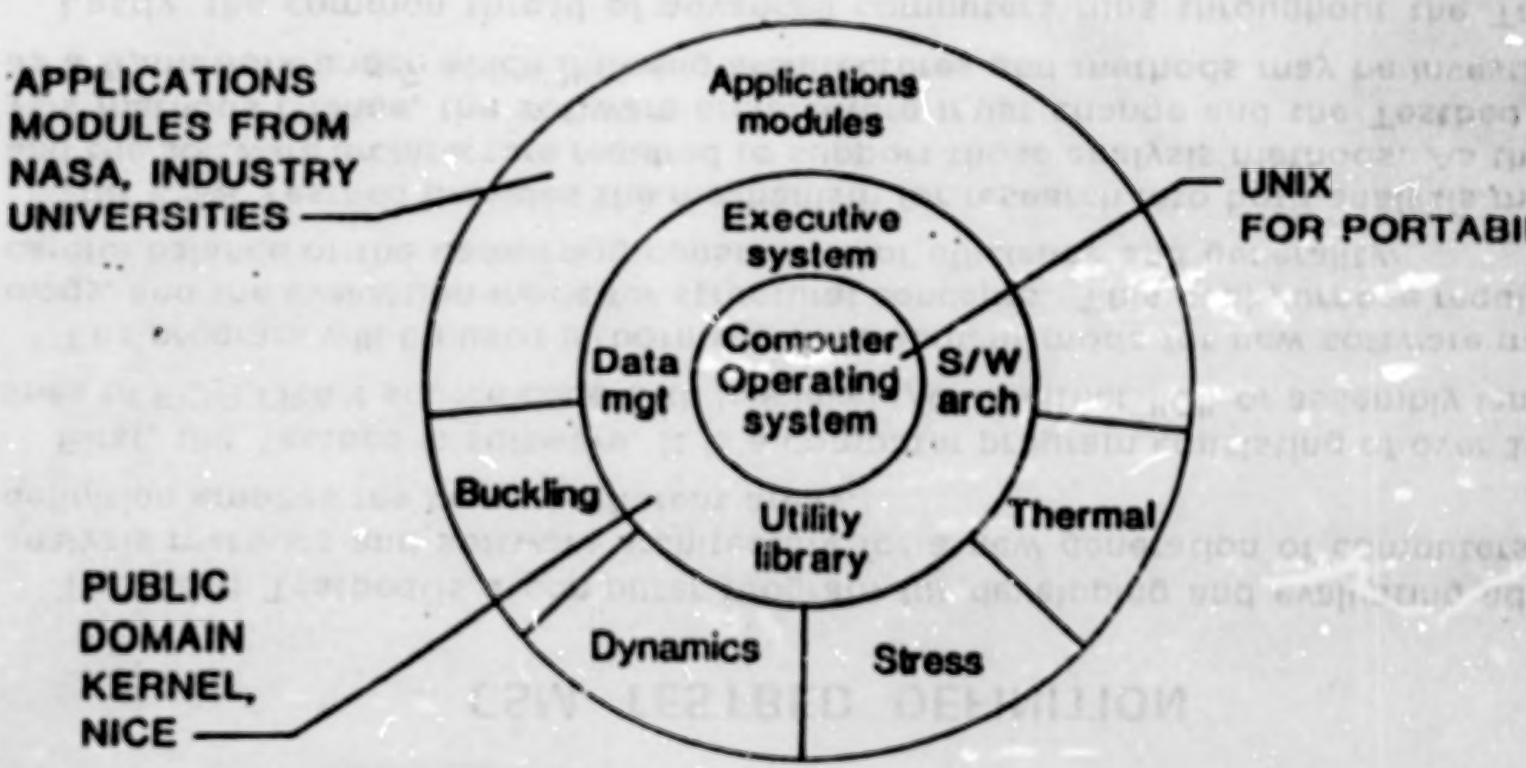
The program will be used in both the development mode for new software methodology, and the evaluation mode for structural concepts. This dual purpose requires the careful balance of the competing constraints of efficiency and generality.

Our CSM Testbed provides the mechanism for research into both analysis methods and the software architecture required to support those analysis methods. As the analysis methods change, the software architecture must change and the Testbed serves as a framework under which differing architectures and methods may be investigated.

Lastly, the common thread of advanced computers runs throughout the Testbed. The development of advanced computer hardware is continuing to accelerate. New architecture computers implementing extremely sophisticated hardware and software concepts that "revolutionize" scientific computing appear two or three times a month. We must be able to evaluate and use these computer enhancements within the CSM area quickly.

COMMON GENERIC SOFTWARE SYSTEM
LARC TESTBED

- PUBLIC DOMAIN SOFTWARE
- MECHANISM FOR TECHNOLOGY TRANSFER
- INDUSTRY, UNIVERSITY, NASA INVOLVEMENT



COMMON GENERIC SOFTWARE SYSTEM

The CSM Testbed as currently implemented at Langley provides a mechanism for technology transfer through the use of public domain software and involves industry, universities, and NASA in a total integrated program.

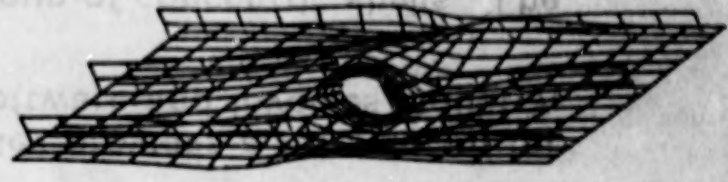
The best way to think of the CSM Testbed is as a group of concentric shells. The inner shell represents the computer operating system. We have concentrated on the UNIX operating system for the Testbed to provide portability. UNIX is the operating system used by all supercomputers and most workstations and is readily available for the range of computers in between. While all versions of UNIX are not directly compatible, they do provide most of the same commands and extensions due to the influence of AT&T. Although we will continue to support the Testbed in VAX VMS environment, new work will be first implemented under UNIX.

The second shell, the executive system shell, consists of the command line interpreter, data management facility, and utility library. This facility was provided initially by the NICE computer code from Lockheed and it has been upgraded to its present state through additional contracts with Lockheed and in-house effort. This shell serves to isolate the outer shell from the computer operating system.

The outer shell consists of the application processors that really do the structural analysis tasks involved in CSM. This outer shell was initially provided by the SPAR system processors and has been augmented through grants, the Lockheed contract and in-house effort.

This Testbed forms a basis on which we can continue the development of structural analysis methodology and apply it to current problems in a way previously not available.

BUCKLING ANALYSIS USING THE CSM TESTBED

PROCESSOR	FUNCTION	RESULT
TAB CSM1★ LAU★ ELD AUS	MODEL GENERATION	 UNDEFORMED SHAPE
RSEQ★ E EKS TOPO K	FORM AND ASSEMBLE STIFFNESS MATRICES	
INV SSOL	FACTOR AND SOLVE	
GSF PSF	STRESS RECOVERY	LINEAR SOLUTION
KG EIG	BUCKLING ANALYSIS	 BUCKLING MODE

★NEW PROCESSORS DEVELOPED FOR THE TESTBED

BUCKLING ANALYSIS USING THE CSM TESTBED

An example of a problem the Testbed can be used to solve is the buckling analysis of the CSM Focus problem. The desired result of the analysis can be seen in the lower right corner plot of the buckling mode. Intermediate results are used to check the progress of the solution. For example, the undeformed shape in the upper right corner is used to check the output of the model generation phase.

The model generation phase of the solution involves the processors TAB, CSM1, LAU, ELD, and AUS. The processors CSM1 and LAU were generated in-house in order to routinely generate this type of structure and to describe the laminations of a composite material.

The linear solution, center right, results from the continued analysis phases required to form and assemble the stiffness matrices with processors RSEQ, E, EKS, TOPO, and K. The RSEQ renumbering processor was developed to reduce the computational overhead associated with the system of equations. Factoring and solving these matrices are accomplished with the processors INV and SSOL. Actual stresses are recovered through the processors GSF and PSF.

The KG and EIG processors are used to complete the buckling analysis computations. The graphic output in this case was provided by PLTA and PLTB.

The software model implemented here is one of a series of computational processors controlled through the use of a higher level language with data management provided through an efficient data base manager.

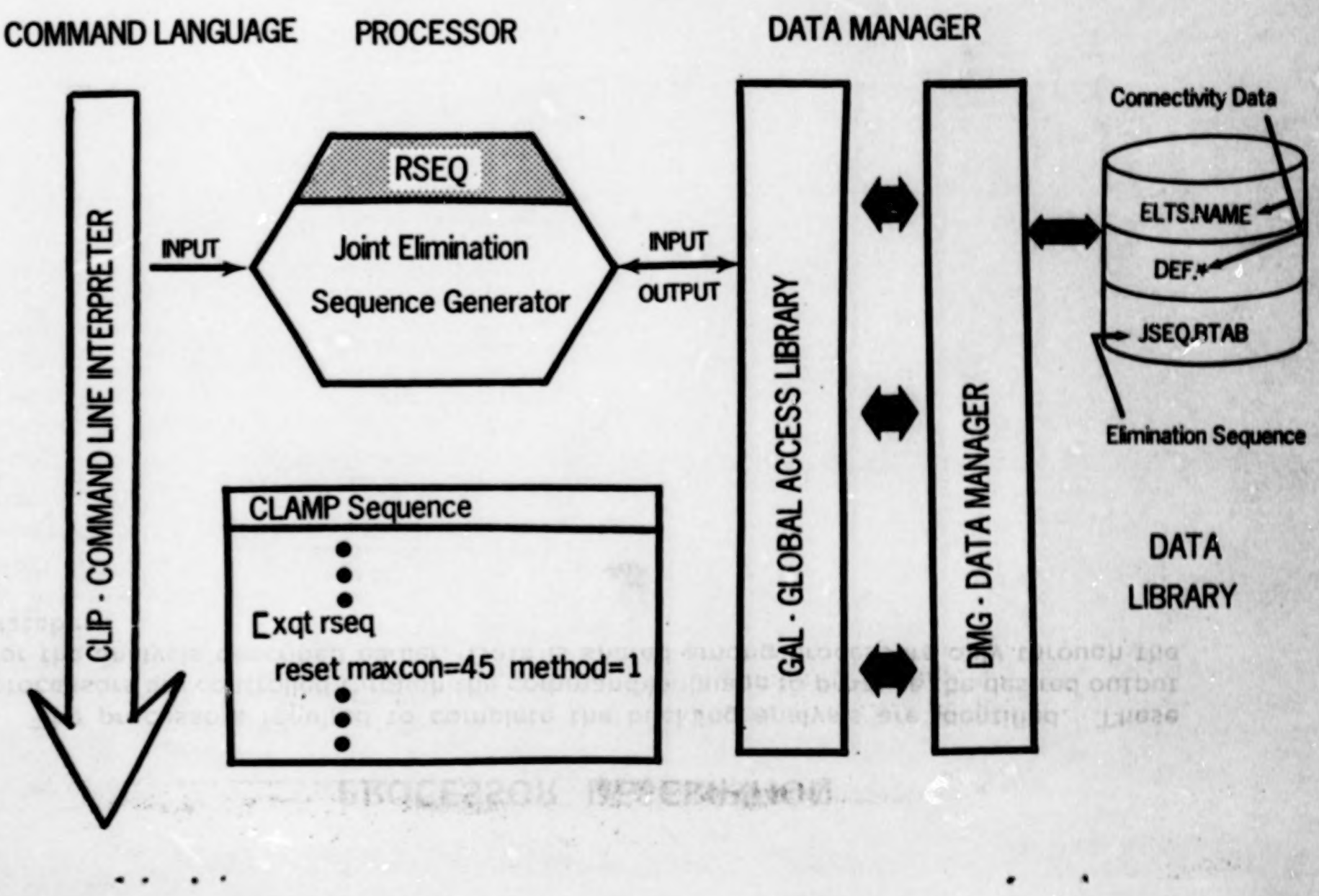
PROCESSOR DESCRIPTION

TAB	Basic Table Input
CSM1	Focus Problem Mesh Generation
LAU	Laminate Analysis Utility
ELD	Element Definition Processor
AUS	Arithmetic Utility System
RSEQ	Renumbering Strategies
E	E-State Initiation
TOPO	Element Topology Analyzer
K	System Stiffness Matrix Assembler
INV	SPAR Format Matrix Decomposition
SSOL	Static Solution Generation
GSF	Stress Data Generation
PSF	Stress Table Printer
KG	Geometric Stiffness Matrix Assembler
EIG	Sparse Matrix Eigensolver

PROCESSOR DESCRIPTION

The processors required to complete the buckling analysis are identified. These processors are controlled through the command language to produce the desired output for the analysis described earlier. Data is shared among processors only through the database.

PROCESSOR-ARCHITECTURE INTERACTION



PROCESSOR-ARCHITECTURE INTERACTION

Looking down one level from the buckling analysis level of Testbed application to the processor level we see the basic components of a modern structural analysis program. The processor, an independent analysis capability, is called into execution by the command language interpreter(CLIP). For example the functional processor RSEQ is initiated through the CLAMP(Command Language for Applied Mechanics) command "[xqt rseq". The processor uses CLIP input processing utilities to read and interpret the input parameters maxcon and method in order to correctly reset the default parameters.

RSEQ then reads the connectivity data from the data library through the global access library data base manager, GAL. GAL in turn uses a lower level I/O manager DMG which interfaces with the lower level "C" or assembly language routines to efficiently perform the I/O functions. This multi-level data base manager provides for an efficient machine independent capability by isolating machine dependent functions to the lowest level possible, and provides the most efficient machine interface possible at that level.

The output of the RSEQ processor is the dataset JSEQ.BTAB. This elimination sequence dataset determines the order of joint elimination in the solution phase based on the user specified method and can result in significant savings.

Each processor can be modified or replaced independently provided that the output data that would be created by that processor is the same as if it were created through another means.

CLIP COMMAND INTERPRETER

Function

- Execute CLAMP Directives to Control Processor Execution
- Provide a Data Description Language for Processor Input

CLAMP Directives

- Interface with the Global Data Manager
  (*OPEN, *CLOSE, *TOC)
- Provide Command Procedure Management
  (*PROCEDURE, *CALL)
- Command Processing Sequence Control
  (*IF, *DO, *JUMP, *RETURN)
- Macrosymbol Directives
  (*DEFINE, *SHOW MACRO, *UNDEFINE)
- SuperClip Directives
  (*RUN, *STOP)
- Data Transfer Directives
  (*LOAD, *UNLOAD)
- General Directives
  (*HELP, *SHOW, *SET, *ADD, *DUMP)

CLIP COMMAND INTERPRETER

The CLIP command interpreter provides two basic functions for the processor. It controls the flow of input records to the executing processor via CLAMP directives, and it provides utilities to expand, construct, and parse processor input.

The CLAMP directives are broken down into seven general areas:

1. **Interface with the Global Data Manager**
2. **Provide command procedure management**
3. **Command processing sequence control**
4. **Macrosymbol directives**
5. **Superclip directives**
6. **Data transfer directives**
7. **General directives**

TESTBED DATA MANAGEMENT

Features

- Global Data Libraries Contain Data Sets
- Data Sets are Accessed by Name
- Low Level I/O is Very Machine Dependent
- Data Storage is Addressed by Word
- Implementation is Hierarchical

Problems

- Data Set Names are Accessed Sequentially
- Management of Memory Resident Data

TESTBED DATA MANAGEMENT

Testbed data management is based on a hierarchic organization with datasets grouped into global data libraries which are currently implemented as files. Within the global data libraries (files) datasets are accessed by name by processors. Efficiency is important due to the size of the datasets that are involved in a state-of-the-art structural problem, and for this reason we resort to low level I/O that is written in assembly language, or "C" and is very machine dependent. We implement a word addressable scheme for managing data to simplify data address calculations. This results in a data handling scheme that provides the necessary efficiency without sacrificing simplicity.

The current implementation of the data management system is not without problems. For example, dataset names are stored in a directory structure called a TOC (table of contents) and are searched sequentially. This has resulted in excessive time when the number of datasets gets large. We also maintain all data in the data management system in the form of files. We do not allow for memory resident data. In this day of computers with large memories, if we could avoid using disk we would decrease the time required for an analysis. We are working to resolve both of these problems at the current time and some partial results will be shown.

TESTBED ARCHITECTURE IMPROVEMENTS

- DATA MANAGER

  PERFORMANCE ON TRANSIENT RESPONSE EXAMPLE

VERSION	CPU SEC.
1986	1120
1987	462

- COMMAND LANGUAGE

  PERFORMANCE ON 10000 TRIP LOOP

VERSION	CPU SEC.
1986	681
1987	436

TESTBED ARCHITECTURE IMPROVEMENTS

Projects are underway in the two areas of executive control, data management and control language. The overhead of the sequential TOC search was eliminated through the use of a hash table and embedded linked lists. This resulted in a speedup of 2 1/2 for a transient response example problem. CLIP, on the other hand, presents a more difficult challenge. In simple analysis procedures the overhead presented by CLIP is not extreme; however, as the complexity of the procedures increases, we see a disproportionate increase in the CLIP overhead. Preliminary work on CLIP has resulted in a decrease in execution overhead but more work will have to be done in this area if we are to provide the ability to develop complex algorithms entirely within the control language.

CSM TESTBED SOURCE CODE CONTROL
(CSM uVAX/ULTRIX)

- **MASTER SOURCE CODE (MSC) FILES FOR NICE AND SPAR**

  Multiple machine versions in single file
  VAX/VMS, VAX/ULTRIX, CRAY/UNICOS, SUN/UNIX

- **"TOOLS" for extracting compatible source code for target systems**
  MAX, INCLUDE

- **"MAKEFILES" for building executable program on target systems**

- **Procedures for distributing latest version of MSC, TOOLS, and MAKEFILES for target systems**

- **RCS (Revision Control System)**

  UNIX utility for maintaining history of modifications to testbed code

CSM TESTBED SOURCE CODE CONTROL

The CSM Testbed is available for many different computers and many different operating systems. If we are to be able to move the technology represented by the Testbed rapidly to a new architecture computer we must have efficient procedures to work with the source code. To accomplish this a group of "tools" has been assembled. We use the pre-compilers developed by Carlos Felippa, MAX and INCLUDE, to provide for extracting compatible source code for target machines. We use makefiles for building executable programs on target machines. We use the Revision Control System (RCS) provided with the UNIX operating system to maintain a history of modifications to the master source code. In addition, we have developed distribution procedures that allow for rapid system updates.

If we are to have a number of simultaneous developers we must maintain control of the Testbed source code and by maintaining strict control of the source code development we are able to transfer new capabilities faster.

CSM Testbed Documentation Set

Proposed Manuals

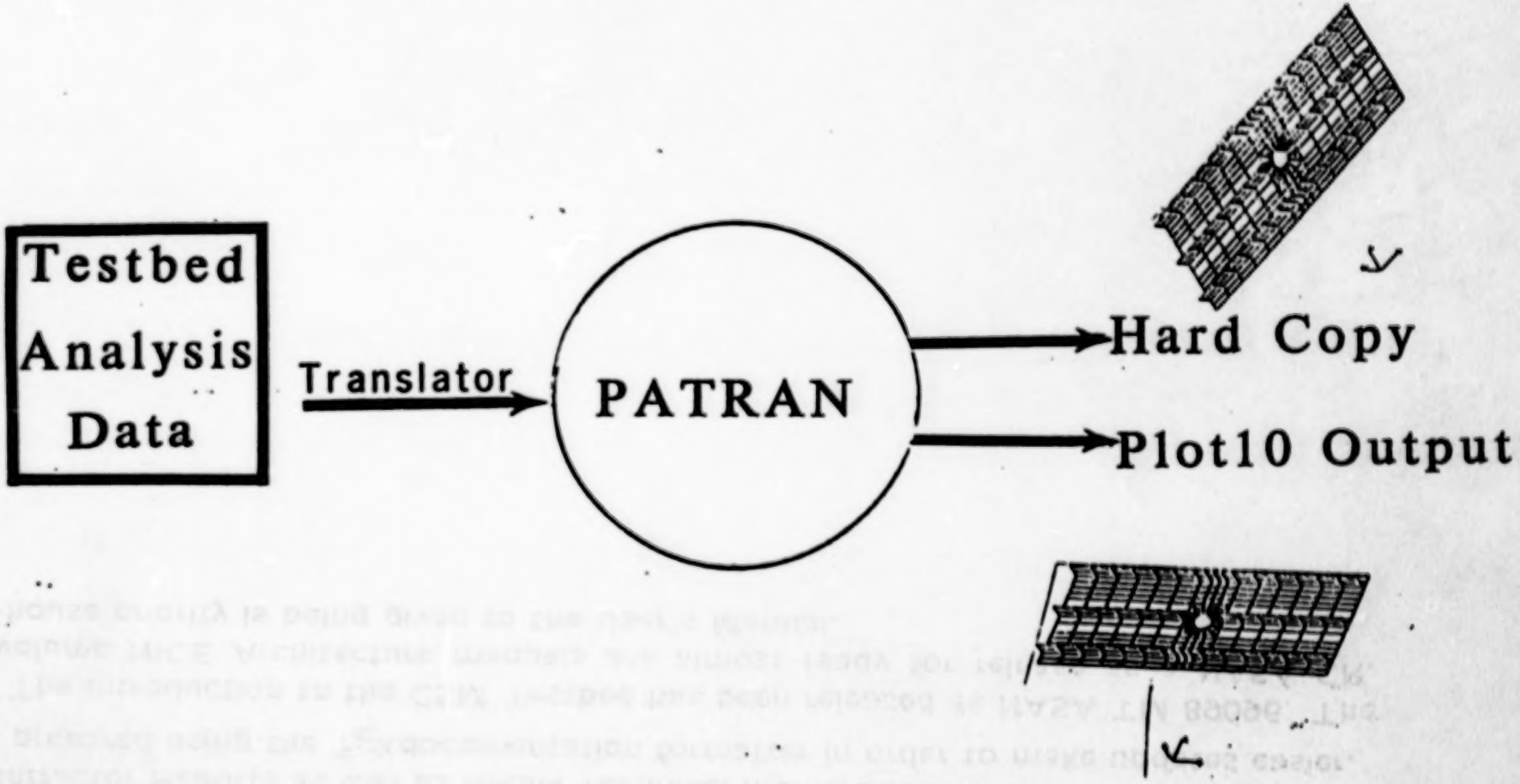
1. User's Manual (50%)
2. Theory Manual (5%)
3. Demo Manual (updated as available)
4. Programmer's Manual (0%)
5. Architecture Manuals (NASA CR, 5 volumes) (99%)
6. Data Library Description (90%)
7. Introduction (NASA TM 89096)
8. Documentation and Programming Standards Manual (0%)
9. Software Tools (NASA CR) (99%)

CSM TESTBED DOCUMENTATION SET

The nine manuals proposed in this documentation set are a priority item for the CSM Testbed development team. We are collecting and publishing material through NASA Contractor Reports as well as NASA Technical Memorandums. All documentation will be prepared using the TEXdocumentation formatter in order to make updates easier.

The introduction to the CSM Testbed has been released as NASA TM 89096. The 5-volume NICE Architecture manuals are almost ready for release as a NASA CR. In-house priority is being given to the User's Manual.

TESTBED GRAPHICS POSTPROCESSING



TESTBED GRAPHICS POSTPROCESSING

We are using the PATRAN system to provide output postprocessing for the CSM Testbed. In order to interface this product with the Testbed a translator was developed to convert the output of the Testbed for use by PATRAN. In addition, the ability to get simple line drawings out of PATRAN required an interface for PATRAN to be developed to allow a file compatible with the Tektronix PLOT10 output to be generated. These two translators were developed in-house and are now available. Hard copy output from PATRAN can now be either through the hard copy device on the graphics display device, or through simple line drawings output and processed through the PLOT10 translator.

Coupling PATRAN with the Testbed gives us the ability to delay a decision on the in-house development of a graphics code. At the present we are concentrating our effort in the development of code that would actually perform the structural analysis in a truly dynamic computational environment. We will review the decision on graphics at a later time.

TESTBED DEVELOPMENT TIMETABLE

1987

- CLEAN UP EXISTING TESTBED
- IMPROVE EFFICIENCY AND EXTENDABILITY
- ADD MODEST NUMBER OF NEW CAPABILITIES
- IMPROVE DOCUMENTATION
- DEVELOP COMPUTER INDEPENDENT CODE MANAGEMENT TECHNIQUES
- IMPLEMENT GRAPHICS POSTPROCESSING (PATRAN)

1988

- IMPLEMENT GRAPHICS PREPROCESSING (PATRAN)
- USE FOR EVALUATING STRUCTURAL ANALYSIS AND COMPUTATIONAL METHODS, DATA HANDLING TECHNIQUES, HIGHER-ORDER LANGUAGES, AND ADVANCED ARCHITECTURE CONCEPTS
- BEGIN TO MODIFY ARCHITECTURE TO EXPLOIT PARALLELISM
- USE FOR SOLVING CHALLENGING PROBLEMS

1989

- MOVE AGGRESSIVELY TOWARD ARCHITECTURE FOR EXPLOITING PARALLEL COMPUTERS
- PROVIDE POWERFUL, EASY-TO-USE TESTBED

TESTBED TIMETABLE

The timetable for the CSM Testbed group is presented for the last year, the next year, and through 1989.

In the last year we concentrated on making the existing Testbed more usable. We put a large effort in cleaning up the Testbed and improving the efficiency. We extended the computer environment onto new processors including high end workstations and supercomputers. We added a modest number of new capabilities to the existing SPAR processors and the NICE architecture. We have improved and continue to improve the documentation to allow a growing number of developers to have access to the Testbed. We have developed a comprehensive set of computer independent code management techniques that permit rapid growth while maintaining tight control of the source code. And we have implemented graphic postprocessing through PATRAN.

In the next year we will continue with the items from last year with a stronger emphasis on problem solving. We will implement a graphics preprocessing interface to PATRAN that will provide a greatly improved user interface. We also expect this preprocessor will assist in converting NASTRAN models for use in the Testbed, for example. We intend to use higher level language development tools to produce an improved command language for the Testbed. We will also begin to modify the software architecture to exploit the new parallel computers. And above all we will prove the Testbed's capability for solving new, challenging problems.

As we go farther out two years to 1989 we get more vague about the tactics of Testbed development and concentrate on the overall strategy. We will move toward a software architecture that will exploit powerful parallel computers while continuing to provide a powerful, easy-to-use Testbed.

TESTBED SCHEDULE

(Near Term – Next 12 months)

- COMPLETE DOCUMENTATION CURRENTLY IN PROGRESS
- INSTALL NEW PROCESSOR CAPABILITIES
- INSTALL ARCHITECTURE ENHANCEMENTS
- EXTEND SUPERCLIP TO UNIX
- REPLACE CLIP PARSER WITH IN-HOUSE DEVELOPMENT
- DEVELOP PATRAN PREPROCESSOR
- EXTEND PARALLEL ALGORITHM DEVELOPMENT TO NAS

TESTBED SCHEDULE

Looking at the next year we are expecting to see several milestones that represent real accomplishments in enhancing the usability of the Testbed.

1. Completing the documentation in progress. If the Testbed is to be usable by others besides the core CSM group, we must have good documentation available. The current documentation is usable by our in-house team.

2. New processors are being developed in-house, under the Lockheed contract, and as a result of grants. The Testbed group will be the clearing house for implementing the processors, testing their integration, and distributing them to others involved in the CSM activity.

3. Software Architecture enhancements are being developed in-house and under the Lockheed contract at the present time. The Testbed group will integrate the new enhancements into the Testbed and verify correct operation of a set of representative demonstration problems.

4. NICE, as originally envisioned by its developer Carlos Felippa, was considered to be a "Network of Independent Computational Elements", since such processors would be independently executed programs linked together by a loosely coupled executive system. This Supercilp environment as it is called is currently available only on the VAX under VMS. Within the UNIX environment the Testbed is currently implemented as a single executable file that has all available processors linked into a large macro-processor environment. We intend to implement the VAX type of Supercilp environment under UNIX as a first step toward adding the constructs that will permit parallel execution of the Testbed.

5. The largest single effort the Testbed group has undertaken for the upcoming year is the replacement of the NICE parser with an in-house development. The new parser will use the UNIX based utilities LEX and YAC to replace the ad-hoc generated parser in NICE. This will allow us to add new constructs to the command language quickly and easily through these UNIX table driven utilities.

6. A PATRAN preprocessor will be developed during the next year to provide the much needed user interface.

7. Many algorithms have been developed in-house by the parallel structural methods group. These algorithms for the most part have been developed for a FLEX-32 that the CSM group owns. The Testbed group plans to take the most promising algorithm produced by the parallel group and implement it on the CRAY-2 at Ames and evaluate the results.

CSM TESTBED SUMMARY

DEFINITION

A computer program for developing and evaluating advanced analysis methods and software architecture for a new generation of computers.

STATUS

- Work is progressing according to schedule
- Updated computer code is ready for release 1.1
- Documentation has reached the usable stage
- Enhancements are underway

CSM TESTBED SUMMARY

The CSM Testbed as it now exists is a computer program for developing and evaluating advanced analysis methods and software architecture for a new generation of computers. It works now, and the many enhancements planned provide the needed framework for research in this area.

Work is continuing according to an established schedule and will result in a capability that will advance the state-of-the-art.

The updated Testbed code is ready for release 1.1 with release 1.2 enhancements scheduled in about 3 months.

The documentation has reached the usable stage and will provide the first external documentation through the User's Manual within the next 4 months.

New capabilities are underway that will lead us through the next generation of computers. We intend to make this capability unique in the CSM research environment.

N89
29791
UNCLAS

PRECEDING PAGE BLANK NOT FILMED 419

CSM Testbed Architecture

by
Philip Underwood
Lockheed Palo Alto Research Laboratory
Computational Mechanics Section

Abstract

This presentation on the CSM Testbed Architecture includes the background for the CSM Testbed Architecture and a description of the three architecture related tasks under the NASA/Langley CSM contract (NAS1-18444): 1) Task 2 — Near-Term Enhancements, 2) Task 5 — Matrix Algebra Methods and Utilities, and 3) Task 6 — Data Management for Parallel Computers. For each task the objectives, subtasks chosen to achieve the objectives, and the accomplishments are presented.

For Task 2, Near-Term Enhancements to the CSM Testbed Architecture, the primary objectives are to quickly correct: 1) inefficiency in the GAL data manager, and 2) deficiency in the CLIP-interpreted command language, CLAMP. The corresponding modifications should preserve upward-compatibility to a reasonable extent, while at the same time increasing the flexibility and extensibility of the CSM Testbed.

We have increased the efficiency of GAL by a factor of 2+ and we have made several improvements in CLIP.

For Task 5, Matrix Algebra Methods and Utilities, the goal is to investigate the current capabilities of the CSM Testbed and the required improvements to the CSM Testbed for performing the matrix algebraic operations required for the analysis of present and future CSM focal problems using advanceds methods.

We have made an extensive study of the matrix algebra functions in SPAR that includes documenting these routines and supplying inline comments.

For Task 6, Data Management for Parallel Computers, the goal is to explore and develop concepts for managing structural analysis data on MIMD computers. The ultimate objective is to develop and implement parallel I/O in the GAL database manager used in the CSM Testbed.

We have designed a parallel I/O system based on domain decomposition. Our next step is to implement the design.

In closing we discuss a possible future extention of the CSM Testbed Architecture, a visual user interface. This user interface would increase the effectiveness and efficiency of analysts and developers using the CSM Testbed.

N89-29701

CSM Testbed Architecture

Introduction

Architecture Background

Architecture Tasks

- Near-Term Enhancements
  - Efficiency, Enhancements & Documentation
- Matrix Algebra Methods and Utilities
  - Study, Evaluation & Documentation
- Data Management for Parallel Computers
  - Exploration, Development & Documentation

N88-13127

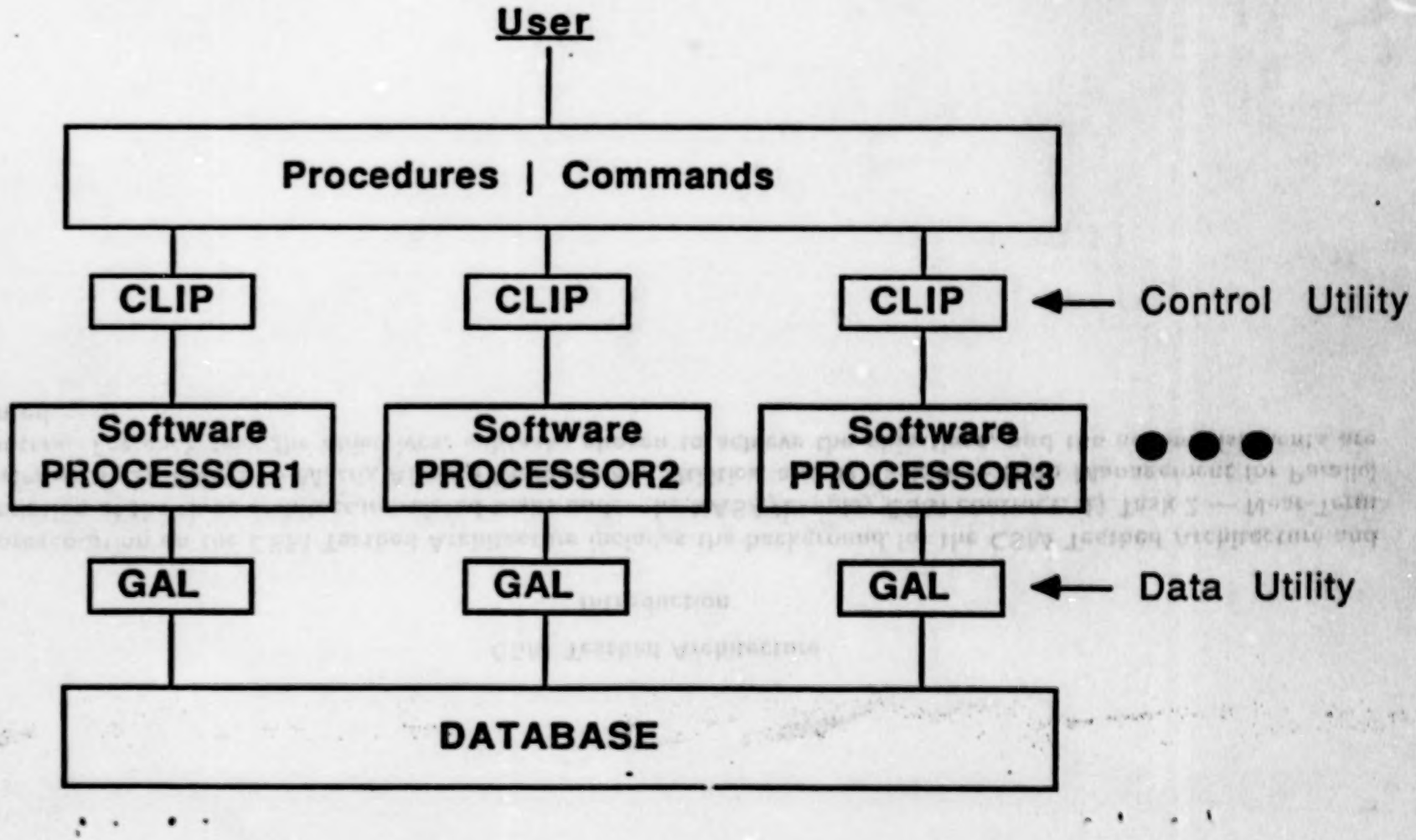
CSM Testbed Architecture

Introduction

This presentation on the CSM Testbed Architecture includes the background for the CSM Testbed Architecture and a description of the three architecture related tasks under the NASA/Langley CSM contract: 1) Task 2 — Near-Term Enhancements, 2) Task 5 — Matrix Algebra Methods and Utilities, and 3) Task 6 — Data Management for Parallel Computers. For each task the objectives, subtasks chosen to achieve the objectives, and the accomplishments are presented.

CSM Testbed Architecture

Architecture Background



CSM Testbed Architecture

Architecture Background

The CSM Testbed Architecture is based on the integrated software system called NICE (Network of Interactive Computational Elements). NICE began as a long-term software-research project in 1980 at the Lockheed Palo Alto Research Laboratory [1],[2]. NICE consists of a generic architecture that implements what computer scientists call a "virtual machine," and of independently-executable, application-specific processors. The architecture consists of three software components that serve all processors: 1) the command language interpreter CLIP, 2) the global database manager GAL, and 3) miscellaneous tools to pre-process, scan and maintain source code.

The processors perform the useful work and communicate through a global database that survives processor execution. A key point is that NICE has no central executive. Each processor, being independently executable, has its own main program and its own command-interpretation shell. Operational compatibility is achieved by replicating two architecture components: the command interpreter CLIP and the data manager GAL, in each processor.

In the original CSM Testbed the processors are from SPAR [3]. They have been augmented by Processors and Procedures developed under the CSM contract to give the CSM Testbed non-linear solution capability.

Near-Term Enhancements

Objectives & Subtasks

Objectives

- Quickly correct inefficiency in the GAL database manager
- Quickly correct deficiency in the CLIP command language interpreter

Subtasks

- GAL-DBM Efficiency
  - Improve TOC Search
  - Investigate Local Memory Management
  - Redesign of GAL Database File Structure
- CLIP Enhancements
  - Clean Up Deficiencies
  - Implement Simple Variable Macrosymbols
  - Enhance Macrosymbol Hashing Techniques
- Improve Documentation
  - Deliver and Present Immediate-Term Enhancements
  - Write Draft NICE Application Programmer's Tutorial
  - Write Draft NICE Quick Reference Document
  - Revise NICE Documentation

Near-Term Enhancements

Objectives & Subtasks

For Task 2, Near-Term Enhancements to the CSM Testbed Architecture, the primary objectives are to quickly correct: 1) inefficiency in the GAL data manager, and 2) deficiency in the CLIP-interpreted command language, CLAMP. The corresponding modifications should preserve upward-compatibility to a reasonable extent, while at the same time increasing the flexibility and extensibility of the CSM Testbed.

The subtasks chosen to achieve the objectives are:

Subtask 1 — GAL-DBM Efficiency: To improve the efficiency of GAL a new TOC (Table of Contents) search algorithm will be implemented. Additionally, local memory management and a redesigned file structure will be investigated as possible ways to increase efficiency.

Subtask 2 — CLIP Enhancements: To improve the functionality and efficiency of CLIP we will clean up some functional deficiencies, implement simple variable macrosymbols to speed up loops, and implement improved hashing of macrosymbols to speed look up of macrosymbols.

Subtask 3 — Improve Documentation: To improve documentation we will write a tutorial, quick reference guide, and document any changes in the architecture.

Near-Term Enhancements

Accomplishments

GAL-DBM: Improve TOC Search

- New Hashing
  - Non-masked Names
  - Masked Names
- Factor of 2+ Speed Up

CLIP: Clean Up Deficiencies

(Improve Functionality, Documented — Not Implemented, ...)

- Repaired Scoping of Local Macrosymbols
- Implemented SCOPE Phrase Qualifier in *DEFINE Directive
- Implemented Macrosymbol ↔ GAL Directives
  - *MAC2GAL or *M2G — Write Macro to GAL
    - ▷ Replaces several directives with one directive
  - *GAL2MAC or *G2M — Read GAL into Macro
    - ▷ Replaces several directives with one directive

CLIP: Implement Simple Variable Macrosymbols

- Work in Progress
  - Modest efficiency gains — 5% maximum

Near-Term Enhancements

Accomplishments

We have implemented a new hashing scheme to improve TOC search in GAL. This produced a factor of 2+ speed up. This is a significant increase in efficiency.

We have modified CLIP: 1) so that the scoping of local macrosymbols is as documented, 2) to activate the SCOPE phrase qualifier in the •DEFINE directive, 3) to implement two new directives, •MAC2GAL and •GAL2MAC, to directly communicate macrosymbol values to the GAL database and GAL database values to macrosymbols, and 4) to affect small changes to the operation of CLIP to support the applications and methods tasks.

We have partially implemented the simple variable macrosymbol. Initial testing indicates this feature will not increase efficiency.

Near-Term Enhancements

Timing Results (Sun3/160)

Empty do-loop

```
*do $i = 1,1000
*enddo
```

Code Version	CPU-secs.
CSM-Baseline *	41.940
CSM-Simple Vars.	40.300
CSM-Simple Vars.+	39.460

Do-loop with 1 card image

```
*do $i = 1,1000
 <$i>
*enddo
```

Code Version	CPU-secs.
CSM-Baseline	50.700
CSM-Simple Vars.	84.860
NICE-Baseline †	75.200
NICE-Compiled	15.520

Do-loop with multiple card images and directives. (100 times thru loop)

Code Version	CPU-secs.
CSM-Baseline	132.980
NICE-Baseline	131.320
NICE-Compiled	17.100

* CSM-Baseline is NICE with Boulder modifications

† NICE-Baseline is NICE without Boulder modifications

Near-Term Enhancements

Timing Results (Sun3/160)

The first group of timing results show that roughly 5% efficiency improvement can be attained by implementing simple variable macrosymbols for an empty do-loop. This is not a significant improvement, especially if you consider more realistic cases.

The second group of timing results are performed for the simpliest realistic case. Here we see the simple variable macrosymbol implementation (CSM-Simple Vars.) falls down because the symbol $i is no longer stored in card image form and must be converted at every iteration. Please note that, this time represents a quick first attempt, we should be able to improve on this. However, the simple variable macrosymbol implementation will probably only be able to match the baseline results at best. Comparing CSM-Baseline with NICE-Baseline shows the effect of changes made to the CSM Testbed at Boulder in May, 1987. For this simple do-loop the efficiency is significantly improved.

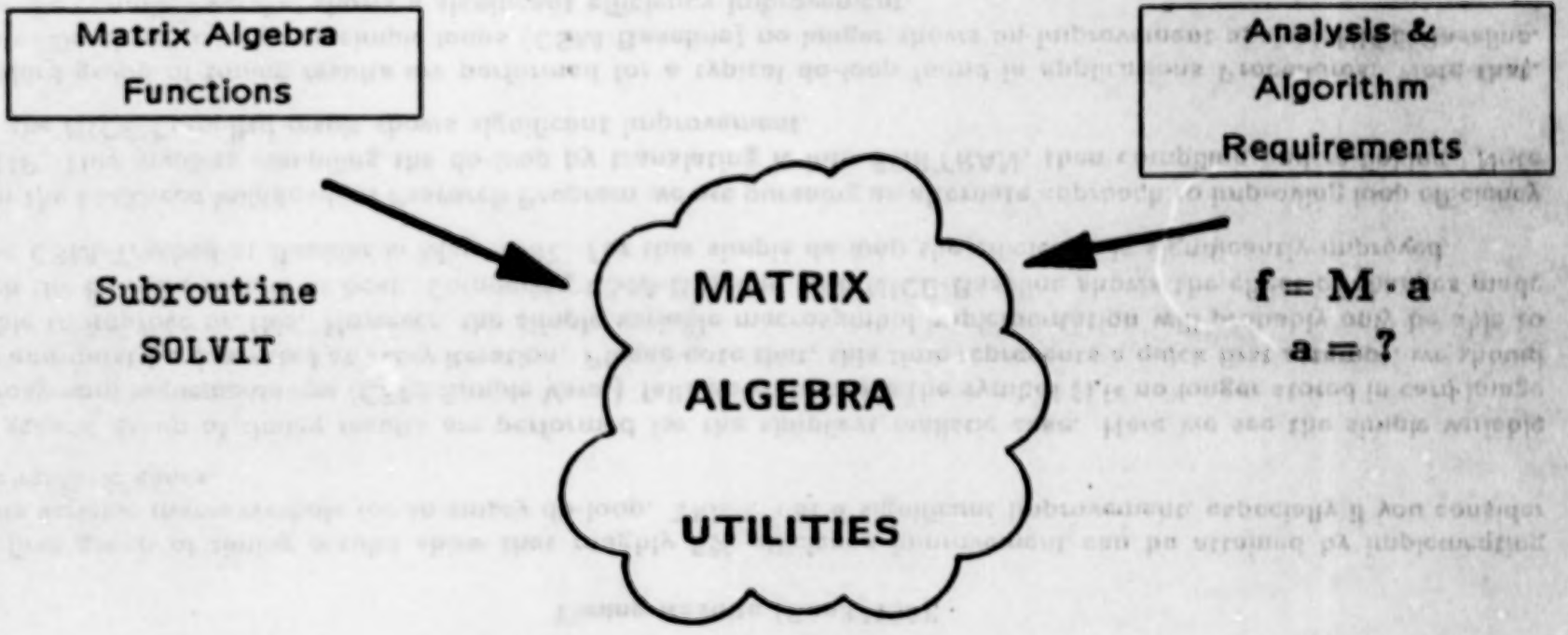
Under the Lockheed Independent Research Program we are pursuing an alternate approach to improving loop efficiency in CLIP. This involves compiling the do-loop by translating it into FORTRAN, then compiling and re-linking. Note that, the NICE-Compiled result shows significant improvement.

The third group of timing results are performed for a typical do-loop found in applications Procedures. Note that, the modifications to enhance simple loops (CSM-Baseline) no longer shows an improvement against NICE-Baseline. Again the compiled version shows a significant efficiency improvement.

The NICE-Compiled design is still undergoing tests and there is much work to be done before it would be ready for implementation in a production environment. However, it appears to be a fruitful direction for research and development.

Matrix Algebra Methods and Utilities

Lockheed Palo Alto *CSM Testbed Architecture*



ORIGINAL PAGE IS OF POOR QUALITY

Matrix Algebra Methods and Utilities

For Task 5, Matrix Algebra Methods and Utilities, the goal is to investigate the current capabilities of the CSM Testbed and the required improvements to the CSM Testbed for performing the matrix algebraic operations required for the analysis of present and future CSM focal problems using advanceds methods.

This initial effort includes a study of matrix algebra routines available in SPAR[3], an evaluation of matrix data structures for advanced algorithms, and documentation.

Matrix Algebra Methods and Utilities

Subtask 1

Lockheed Palo Alto *CSM Testbed Architecture*

Study SPAR Formatted Matrix Data Structures and Processors

- **Details of the data structures.**
- **Interdependence of data structures.**
- **Data-blocking logic in the data structures.**
- **Processor data flow and architecture.**
- **Processor algorithmic logic flow.**
- **Documentation**

Matrix Algebra Methods and Utilities

Subtask 1

Under this subtask the current sparse matrix data structures and the Processors that use them in the CSM Testbed will be studied to determine and document: 1) details of the data structures, 2) interdependence of data structures (e.g., K and KMAP), 3) data-blocking logic in the data structures, 4) Processor data flow and architecture, and 5) Processor algorithmic logic flow.

Inline documentation of SPAR matrix algebraic functions in Processors TOPO, K, INV, SSOL, EIG, and AUS will be provided. Enhancements to the documentation for the current SPAR matrix data structures (e.g., K, KMAP, AMAP, etc.) will also be provided.

Matrix Algebra Methods and Utilities

Subtask 1 — Progress

Processor	Data Structures Decoded	Source Code Deciphered	Source Code Commented	Documentation Written
TOPO	✓	✓	✓	
K	✓	✓	✓	
INV	✓	✓	✓	
SSOL	✓	✓	In Progress	
AUS (SUM PROD)	✓	In Progress	In Progress	
EIG	✓	✓	In Progress	

ORIGINAL PAGE IS OF POOR QUALITY

Matrix Algebra Methods and Utilities

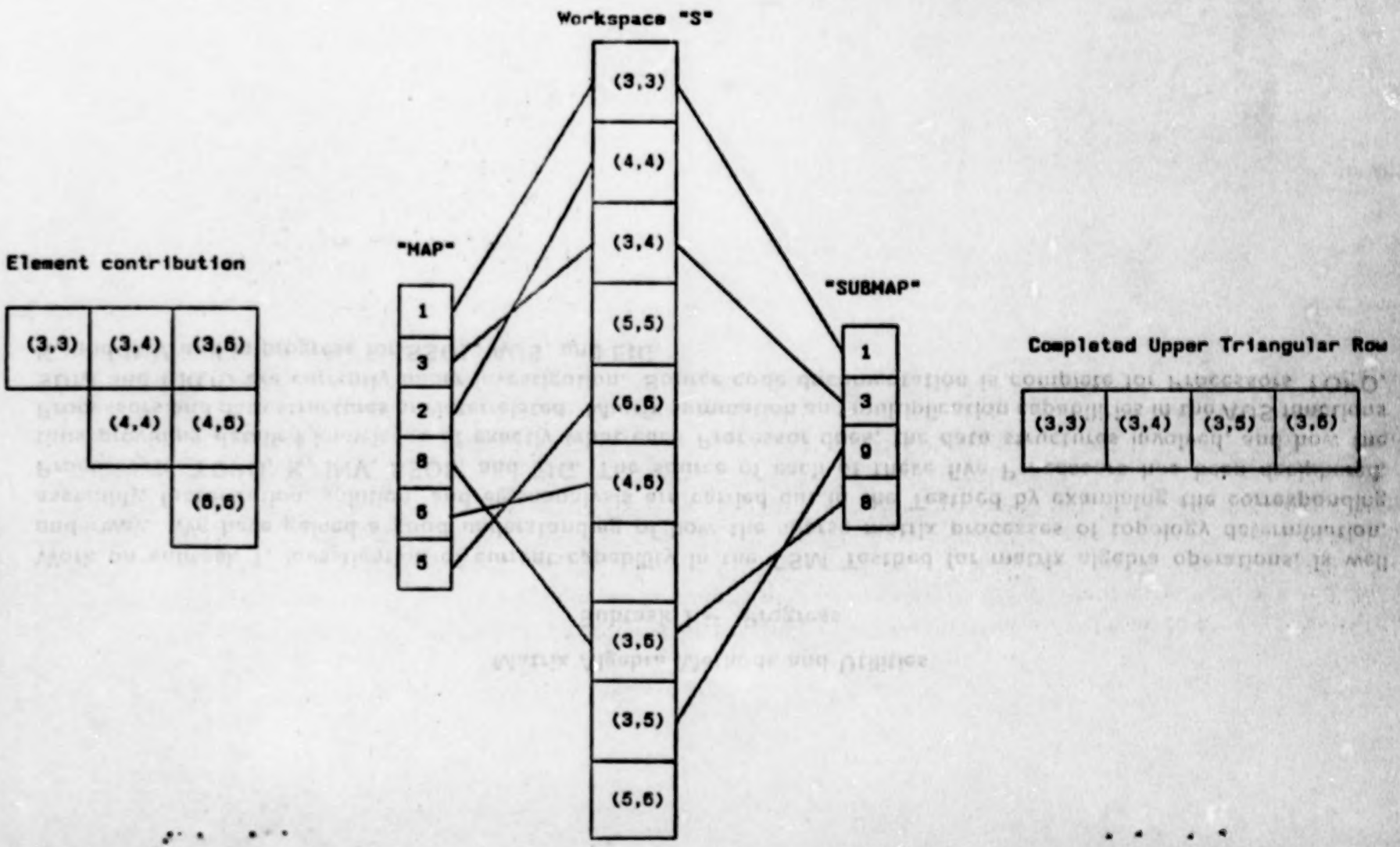
Subtask 1 — Progress

Work on subtask 1, investigation of current capability in the CSM Testbed for matrix algebra operations, is well underway. We have gained a good understanding of how the sparse matrix processes of topology determination, assembly, factorization, solution, and eigenanalysis are carried out in the Testbed by examining the corresponding Processors: TOPO, K, INV, SSOL, and EIG. The source of each of these five Processors has been deciphered, thus providing detailed knowledge of exactly what each Processor does, the data structures involved, and how the Processors and data structures are interrelated. Matrix summation and multiplication capabilities in the AUS functions SUM and PROD are currently under investigation. Source code documentation is complete for Processors TOPO, K, and INV and in progress for SSOL, AUS, and EIG.

Matrix Algebra Methods and Utilities

Subtask 1 — KMAP Use Example

Lockheed Palo Alto *CSM Testbed Architecture*



ORIGINAL PAGE IS
OF POOR QUALITY

Matrix Algebra Methods and Utilities

Subtask 1 — KMAP Use Example

The KMAP dataset may be thought of as a road map to be used during the assembling of the system stiffness matrix. (The same map is also used to assemble the geometric stiffness and the consistent mass matrices.) KMAP contains explicit instructions for element assembly: when to assemble an element, where the element is located, and where to assemble it into the local workspace. It also contains instructions for the archival of completed upper triangular rows from the workspace: when the row is finished, what non-zero submatrices exist for the row, and where each of these submatrices is stored in the workspace.

The example shows what happens during the assembly of the last element attached to joint 3 in a sample problem. The "MAP" for this element contains a pointer into the assembly workspace area, "S," for each elemental submatrix. Once this element's contribution has been assembled, the upper triangular row of the system stiffness corresponding to joint 3 is complete.

"CONECT" is a list of the joints connected to joint 3. This list corresponds to the non-zero submatrices in this joint's row of the system stiffness matrix (except for the diagonal submatrix, which is implicitly included). "SUBMAP" contains a pointer into the assembly workspace for each submatrix to be written out for this joint. Once these submatrices are written out, their space in the workspace is zeroed out and returned to the available pool.

All of the work in figuring out this road map has been done in Processor TOPO. Processors K, KG, and M simply follow the map. Note that, KMAP is only used for assembly, it is not used to interpret an existing SPAR formatted matrix.

Matrix Algebra Methods and Utilities

Subtask 2

Lockheed Palo Alto *CSM Testbed Architecture*

Support CSM Algorithm and Methods Development.

- Dynamics:
  - Multi-body simulation
  - Transient response analysis
  - Joint contact & damping
- Statics:
  - Nonlinear continuation methods
  - Equivalence Transformation
  - Nonlinear substructuring
- Parallel Processing:
  - Mesh partitioning
  - Matrix factor, solve, etc.
  - Vector operations
  - Eigenvalue analysis

Matrix Algebra Methods and Utilities

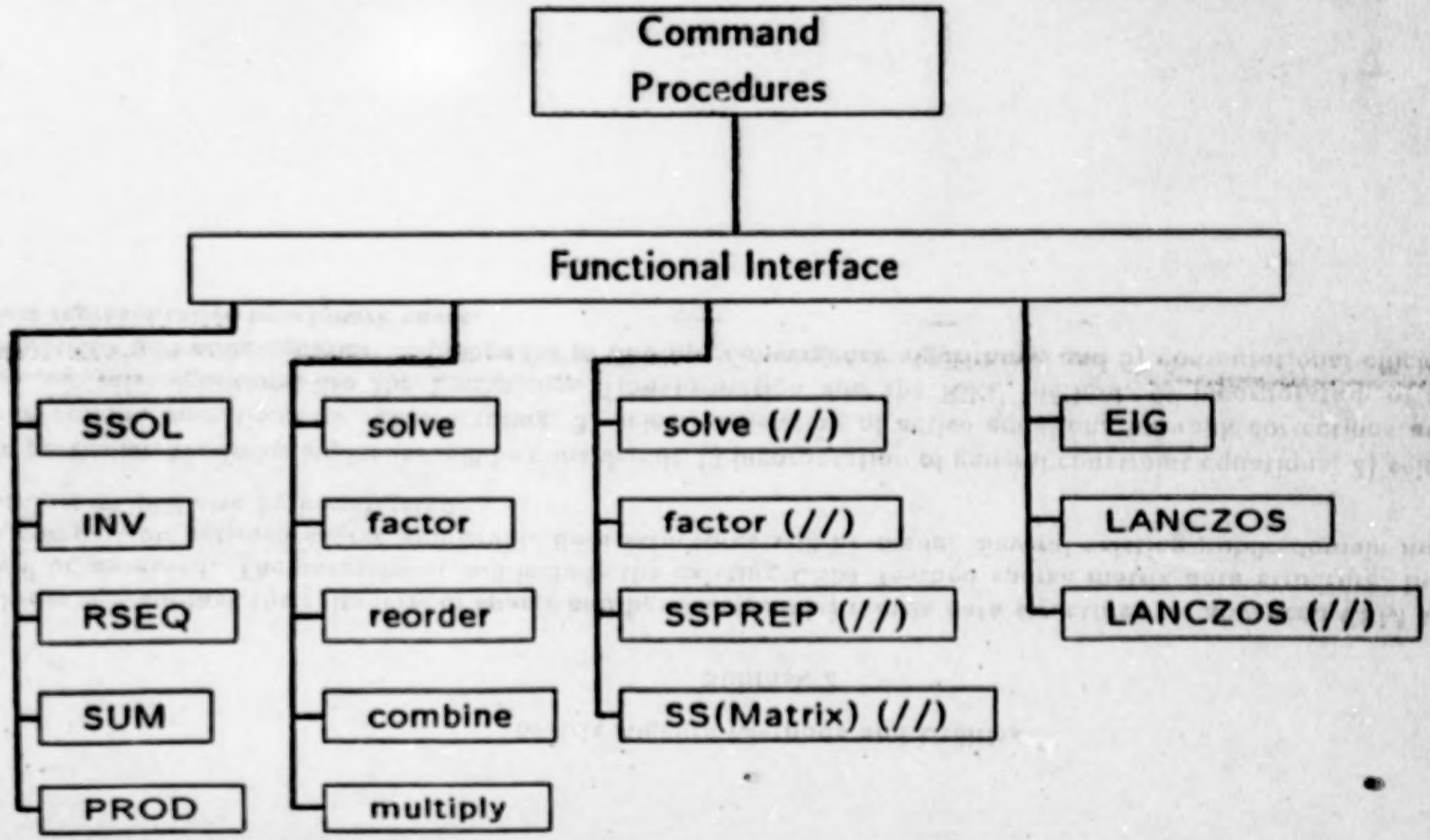
Subtask 2

Under this subtask the suitability of sparse and banded (profile) matrix data structures for advanced CSM algorithms will be assessed. The assessment will include the existing CSM Testbed sparse matrix data structure. In addition, a comparison between sparse and profile data structures will be made. Several existing public-domain matrix data structures will also be investigated.

In particular, the following issues will be considered: 1) incorporation of general constraint equations, 2) selective row and column operations for substructuring, 3) selective deletion of active equations for rank corrections as required by analysis algorithms like the Equivalence Transformation and the Riks' method, 4) incorporation of additional equations and node-equation mappings for p- and hp- convergence algorithms, and 5) computational efficiency for a few representative benchmark cases.

Matrix Algebra Methods and Utilities

The Ultimate Goal



Matrix Algebra Methods and Utilities

The Ultimate Goal

The ultimate goal of this activity is to provide a "complete" set of matrix algebra utilities for developers of new methods and for application to the analysis of advanced CSM focal problems. The matrix algebra utilities should have a breadth from basic operations, such as sum and multiply, to solution packages, such as SSOL and LANCZOS. The implementation of the matrix algebra utilities should be such that the user at the command level can issue commands for a specific operation and have the functional interface determine whether a sequential or parallel execution is appropriate. In addition, the matrix algebra utilities should be available to developers through FORTRAN subroutine calls.

Data Management for Parallel Computers

Goal & Subtasks

Lockheed Palo Alto *CSM Testbed Architecture*

Goal

- Explore and develop concepts for parallel I/O

Subtasks

- Develop data management techniques for MIMD computers.
  - Collaborate with CSM grantees in parallel processing.
  - Must be appropriate for finite element structural analysis.
  - Emphasis on shared memory.
  - Modify CSM Testbed architecture for parallelism.
- Make preliminary calculations using techniques developed.
- Present results.

Data Management for Parallel Computers

Goal & Subtasks

For Task 6, Data Management for Parallel Computers, the goal is to explore and develop concepts for managing structural analysis data on MIMD computers. The ultimate objective is to develop and implement parallel I/O in the GAL database manager used in the CSM Testbed.

To achieve this goal there are three subtasks.

1) In collaboration with Carlos Felippa, Charbel Farhat, Harry Jordan, and other CSM grantees in parallel processing, develop data management techniques for MIMD computers. The techniques will be appropriate for a finite element structural analysis environment. Emphasis will be placed on modifying the CSM Testbed Architecture to account for parallelism.

2) Make preliminary and exploratory calculations using the techniques developed. Make comparisons between data management requirements for shared memory computers and data management requirements for local memory computers.

3) Present results.

Data Management for Parallel Computers

Develop - Appropriate for FE Analysis

Finite Element Analysis

model problem - graphics ||
decompose domain ||
formulate elements ||
assemble ||
solve (direct & iterative) ||
recover stresses ||
update state ||
display results - graphics ||

Generic

- model
- decompose
- formulate (assemble)
- solve
- recover (update)
- display

FUNDAMENTAL DIFFERENCE BETWEEN SEQUENTIAL AND PARALLEL COMPUTING IS

DOMAIN DECOMPOSITION

Data Management for Parallel Computers

Develop - Appropriate for Finite Element Analysis

To develop data management techniques on parallel computers that are appropriate for finite element structural analysis, we first considered how one does finite element analysis on parallel computers. The basic steps in a typical linear, non-linear, static, and dynamic analysis by the finite element method on a parallel computer are: 1) model problem, 2) decompose domain, 3) formulate elements, 4) assemble, 5) solve, 6) recover stresses, 7) update state, and 8) display results [4]. Each step can be computed with parallel algorithms. Thus the finite element method for structural analysis is highly parallelizable.

In a generic sense the finite element structural analysis method on a parallel computer has six steps: 1) model, 2) decompose, 3) formulate, 4) solve, 5) recover, and 6) display. The heaviest arrow indicates the path most used in solving a non-linear or dynamic problem. The heavier arrow indicates that occasionaly the problem changes to the extent that another decomposition is needed. The thinest arrow indicates that at times one looks at the results and starts all over again.

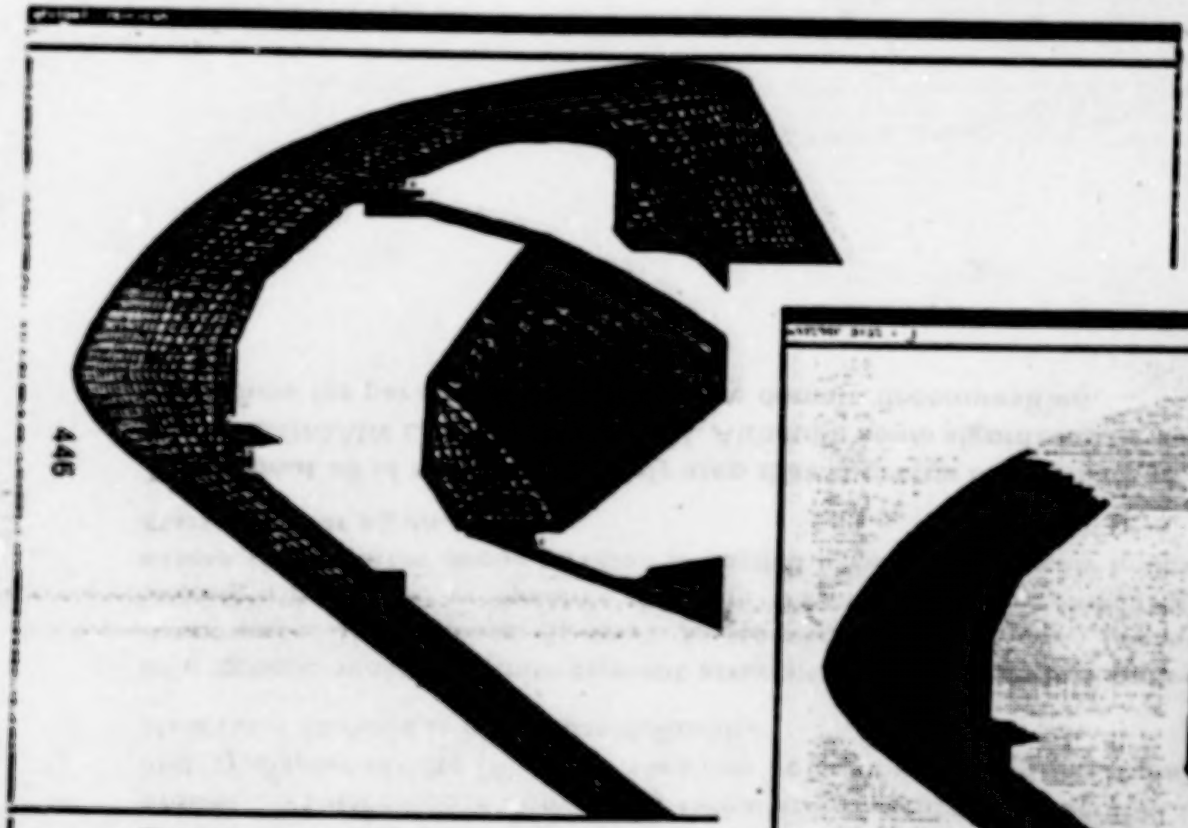
Throughout all of the steps the only step that is unique to the finite element analysis method on parallel computers is that of DOMAIN DECOMPOSITION. Although some algorithms for sequential computers use domain decomposition, algorithms for parallel computers require domain decomposition.

Data Management for Parallel Computers

Domain Decomposition

Lockheed Palo Alto *CSM Testbed Architecture*

ORIGINAL PAGE IS
OF POOR QUALITY



Finite element mesh for nozzle problem (6216 e

Directional Bisection mesh partition for nozzle problem ($N_{proc} = 8$)

Data Management for Parallel Computers

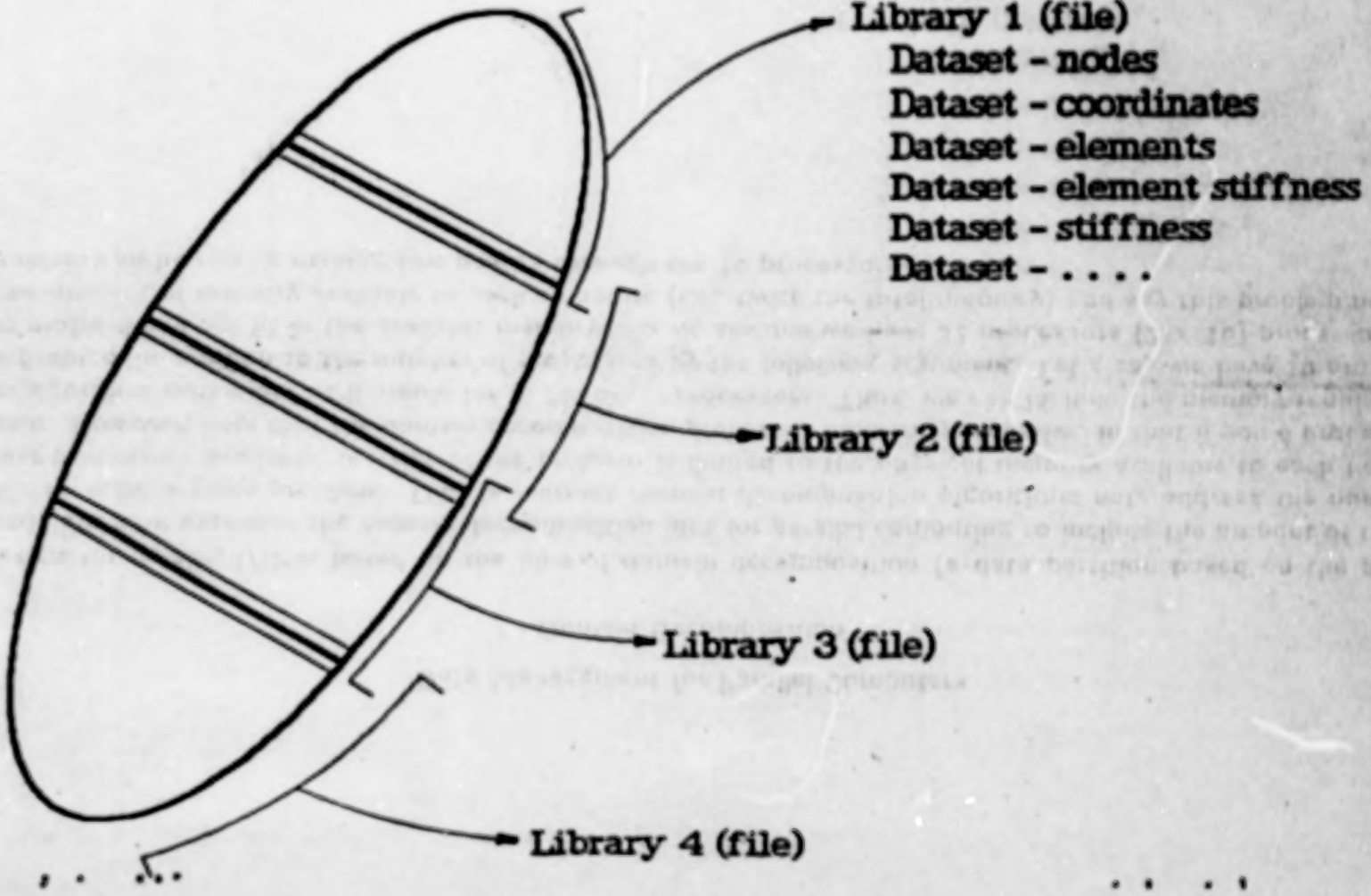
Domain Decomposition

Our design for parallel I/O is based on the idea of domain decomposition (a data partition based on the physical problem). We have extended the domain decomposition idea for parallel computing to include the amount of memory available to solve a given problem. That is, current domain decomposition algorithms only address the number of hardware processors available; the size of the problem is limited to the physical memory available to each hardware processor. However, note that the domain decomposition process is inherently recursive, in that if you 4 processors a parallel algorithm works just as it would for 8, 16, or ... processors. Thus, we can include the memory requirements for the problem in addition to the number of processors by the following argument. Let's say we have 16 processors but the problem will not fit in the available memory. So we assume we have 32 processors (2×16) processors with the same amount of memory available to each as before (i.e., twice the total memory) and say this problem now fits. The problem can be run by making two passes through the 16 processors.

Data Management for Parallel Computers

Domain Decomposition — Data Decomposition

Lockheed Palo Alto *CSM Testbed Architecture*



Data Management for Parallel Computers

Domain Decomposition - Data Decomposition

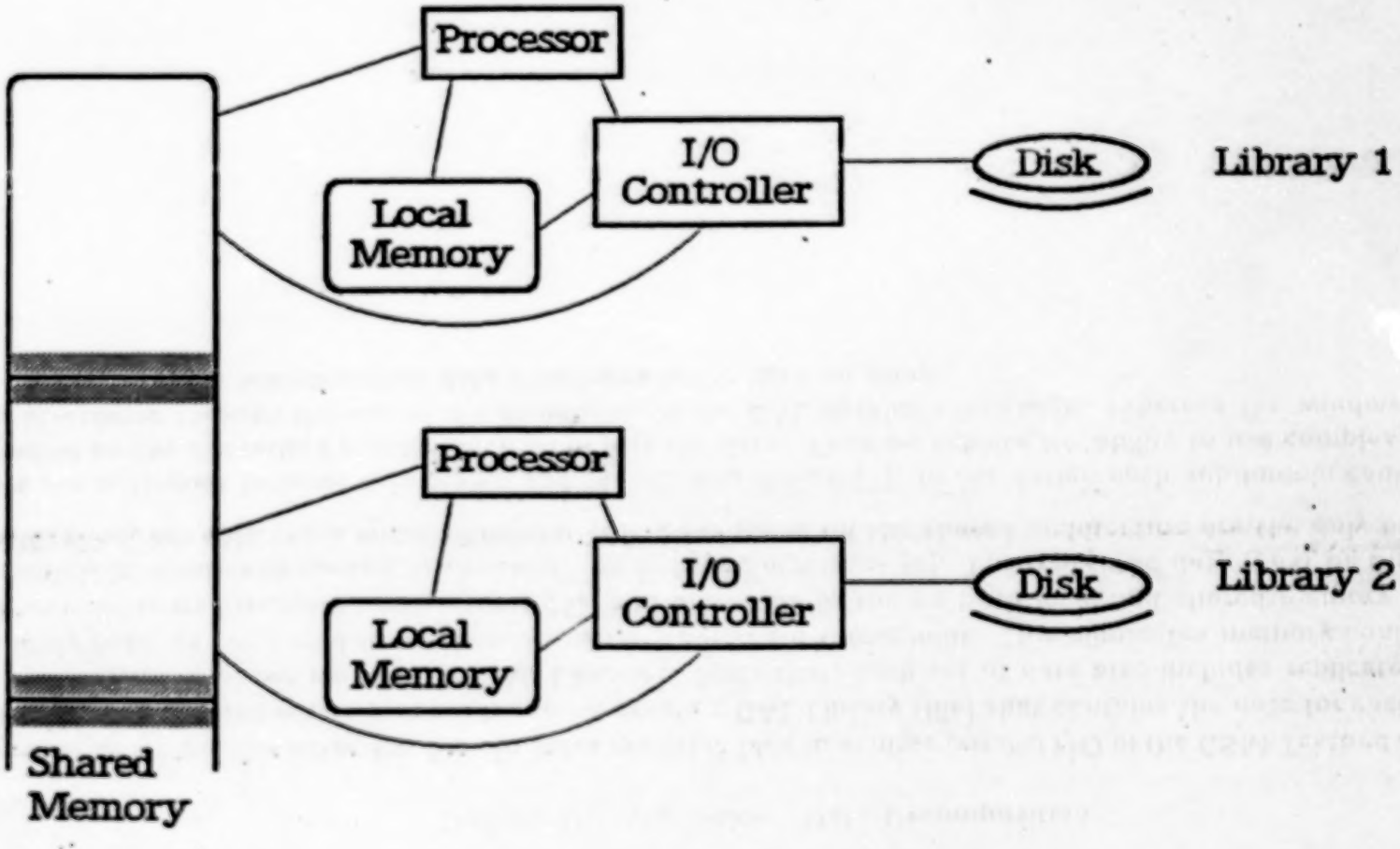
Now how do we use this extended domain decomposition idea to achieve parallel I/O in the CSM Testbed Architecture? As the domain decomposition process is run we create a GAL Library (file) that contains the data for each subdomain. So in the example above we have 32 GAL Libraries. Note that, each set of data also includes replicated subdomain boundary data, as we intend to use a local memory paradigm throughout. This eliminates memory conflict problems that arise in shared memory architectures [5], and allows us to run on both local and shared memory architectures by mimicking a message passing environment on both architectures [6]. The replicated data must be tagged for easy identification, but this and a small additional storage required on the shared architecture are the only overhead.

There are analogues between this design and the window design [7]. In our design each subdomain can be likened to a window on the discretized problem instead of just the data. Thus we achieve the ability to use complex and dynamic data structures through the use of the structures in the GAL database language. Whereas the window design does not appear to easily accommodate data structures other than an array.

Data Management for Parallel Computers

Parallel I/O — Hardware

Lockheed Palo Alto *CSM Testbed Architecture*



replicated data (boundary data)

Data Management for Parallel Computers

Parallel I/O — Hardware

Here we present an idealized picture of a parallel computer architecture that contains shared and local memory, an I/O controller and disk drive for each processor. On each disk we have a GAL Library (actually for our example above each disk would have two GAL Libraries but only one is active). Most likely, in reality, we would have to share I/O controllers and disk drives among all processors or groups of processors. An implementation of our design will have to consider what is actually available to us. However, the Cray2 and the NCUBE have elements of what is shown here. We believe we can map our parallel I/O design onto real architectures.

Assuming we can do this lets walk through the steps in one iteration (cycle) of computation for our example. We initiate the first 16 subdomain problems on the 16 available processors. Each processor reads the data for its subdomain from its GAL Library. This is done in true parallelism for a computer such as illustrated here. Otherwise the hardware operating system with the CSM Testbed Architecture covering the actual hardware system (virtual machine) will handle the I/O requests to make it appear to be parallel I/O. After the data is read each processor does its computations and writes the data back to its GAL Library. Then the second group of 16 subdomains is processed the same way. At this point the replicated boundary data is updated as is normally done in a message-passing local memory architecture (this is equivalent to synchronization on a shared memory architecture). The GAL Libraries are updated during this process but only the boundary data is effected. Now another step can be taken.

Data Management for Parallel Computers

Programming Methodology

Lockheed Palo Alto *CSM Testbed Architecture*

- **Employ a local memory execution model.**
  - **Communicate between processors by passing messages.**
  - **Decompose computational tasks.**
  - **Decompose communications (I/O) tasks.**

    (Do not mix computation and I/O.)
- **Will achieve software that is relatively insensitive to changes in:**
  - **number of processors,**
  - **interconnection structure,**
  - **local or shared memory,**
  - **amount of memory.**

Data Management for Parallel Computers

Programming Methodology

The programming methodology we advocate is the local memory execution model. This means message-passing for communications and decomposition of both computational tasks and I/O tasks. We believe this methodology will achieve software that is relatively insensitive to changes in: the number of processors, the interconnection structure of the computer, local or shared memory, and the amount of memory.

Others have advocated this long before us [5] and [6]. Their influence and results seem to indicated they should be listened to. Also, the message-passing environment is used in PISCES [8] and it is part of the original design for NICE [1]. We plan to stay with success.

Data Management for Parallel Computers

Next Step(s)

- **Implement domain decomposition algorithm in CLAMP.**
  - **generates separate GAL Libraries from model data**
- **Modify GAL to accommodate message passing.**
  - **Send and Receive**
  - **Queue I/O**
- **Try out on Cray2 and/or NCUBE.**

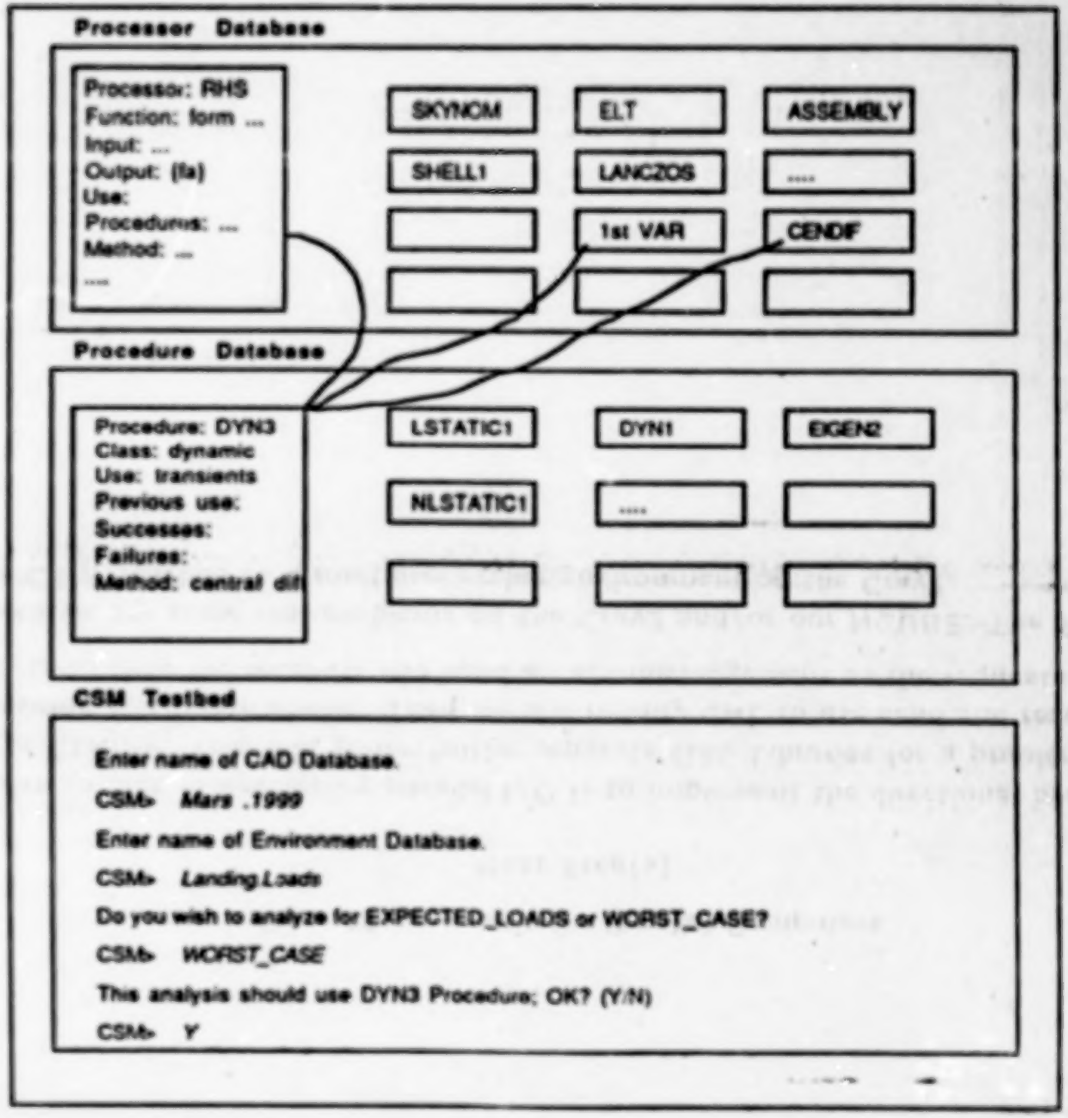
Next Step(s)

The next steps we plan to take in developing parallel I/O is to implement the directional bisection domain decomposition algorithm [9] in CLAMP. This will generate the separate GAL Libraries for a problem such as that shown in the domain decomposition discussion above. Then we will modify GAL to use send and receive commands. Here we will queue I/O requests as they are received and send an acknowledgement as the requests are processed.

Assuming all goes well we try some real problems on the Cray2 and/or our NCUBE. The FORCE [10] may be used to move some of the CSM Testbed to a multiprocessing environment on the Cray2.

CSM Testbed Architecture

Future — User Interface



CSM Testbed Architecture

Future — User Interface

The CSM Testbed Architecture supports a rich and complex set of functions and applications. It is often difficult for a beginner to get started or even for an experienced user to be aware of all that is happening or can be done. A visual user interface would provide the support a user or developer needs to effectively and efficiently use the CSM Testbed.

Powerful engineering workstations are becoming common in the research and development environment and will soon be common in the production analysis environment. With standards such as UNIX, X-Windows, FORTRAN77, and ANSI "C" a portable user interface for the CSM Testbed on engineering workstations is doable and desireable.

References

[1] C. A. Felippa, *Architecture of a distributed processing network for computational mechanics*, **Computers and Structures, 13**, 405-413, 1981

[2] C. A. Felippa and G. M. Stanley, *NICE: A Utility Architecture for Computational Mechanics*, in proceedings of **Finite Element Methods for Nonlinear Problems**, eds., Bergan, Bathe and Wunderlich, Europe-US Symposium, Trondheim, Norway, 1985

[3] W. D. Whetstone, *SPAR Structural Analysis System Reference Manual*, Vol. 1, NASA CR 158970-1, Dec. 1978

[4] C. H. Farhat, *Multiprocessors in Computational Mechanics*, Ph.D. Thesis, UCB/SESM-86/12, Dept. of Civil Engineering, U. of California, Berkeley, CA, Dec. 1986

[5] C. A. R. Hoare, **Communicating Sequential Processes**, §6.3, Prentice/Hall International, Englewood Cliffs, NJ, 1985

[6] S. B. Baden, *Run-Time Partioning of Scientific Continuum Calculations Running on Multiprocessors*, Ph.D. Thesis, LBL-23625, Lawrence Berkeley Laboratory, Berkeley, CA, June 1987

[7] P. Mehrotra and T. W. Pratt, *Language Concepts for Distributed Processing of Large Arrays*, **ACM SIGACT-SIGPLAN Symp. Principles Distributed Computing**, Ottawa, 19-28, Aug. 1982

[8] T. W. Pratt, *The PISCES 2 Parallel Programming Environment*, ICASE Report No. 87-38, NASA Langley Research Center, Hampton, VA, July 1987

[9] B. Nour-Omid and S. N. Glasser, *Recursive Inertial Bisection for Partitioning Finite Element Meshes*, in preparation

[10] H. F. Jordan, M. S. Benten and N. S. Arenstorf, *Force User's Manual*, Dept. of Electrical and Computer Engineering, U. of Colorado, Boulder, CO, October 1986

N89
29792
UNCLAS

STRUCTURES DIVISION
STRUCTURAL MECHANICS BRANCH
NASA Lewis Research Center

COMPUTATIONAL STRUCTURAL MECHANICS
ENGINE STRUCTURES COMPUTATIONAL SIMULATOR

C. C. CHAMIS
NASA LEWIS RESEARCH CENTER

See p 460

NASA WORKSHOP ON COMPUTATIONAL STRUCTURAL MECHANICS
NASA LANGLEY RESEARCH CENTER
NOVEMBER 18-20, 1987

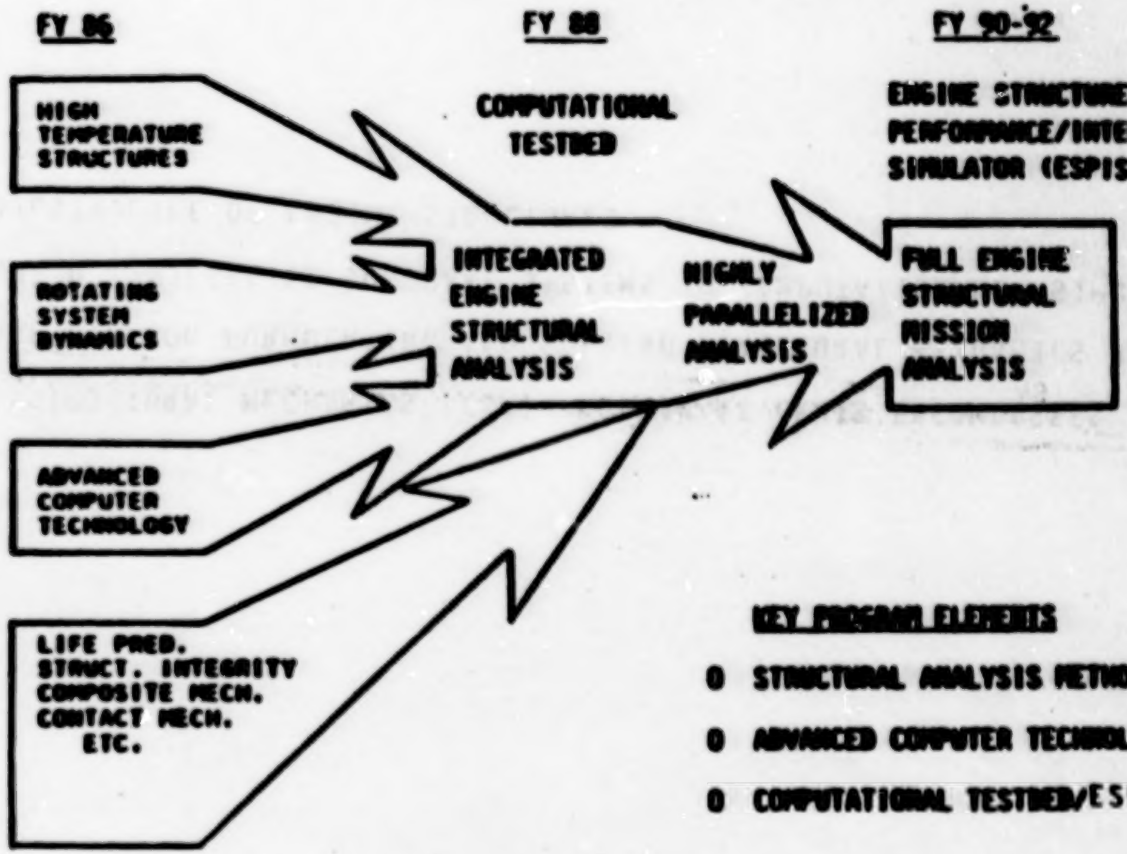
N89-29792
ND 315753
211960
S3-39
27P.

COMPUTATIONAL STRUCTURAL MECHANICS FOR ENGINE STRUCTURES

- Investigate Unique Advantages of Parallel and Multi Processors For:
  - Reformulating/Solving Structural Mechanics
  - Formulating/Solving Multidisciplinary Mechanics
- Develop "Integrated" Structural System Computational Simulators For:
  - Predicting Structural Performance
  - Evaluating Newly Developed Methods
  - Identifying/Prioritizing Improved/Missing Methods Needed

THE COMPUTATIONAL STRUCTURAL MECHANICS (CSM) PROGRAM AT LEWIS ENCOMPASSES (1) FUNDAMENTAL ASPECTS FOR FORMULATING AND SOLVING STRUCTURAL MECHANICS PROBLEMS AND (2) DEVELOPMENT OF INTEGRATED SOFTWARE SYSTEMS TO COMPUTATIONALLY SIMULATE THE PERFORMANCE/DURABILITY/LIFE OF ENGINE STRUCTURES.

COMPUTATIONAL STRUCTURAL MECHANICS



ORIGINAL PAGE IS OF POOR QUALITY

THE GENERAL CONTENT OF THE CSM LEWIS PROGRAM PLAN IS SUMMARIZED IN THE ACCOMPANYING BLOCK DIAGRAM. THE LONG-RANGE OBJECTIVE OF THE PROGRAM IS THE FULL ENGINE STRUCTURAL SIMULATION.

COMPUTATIONAL STRUCTURAL MECHANICS

IDENTIFIED METHODOLOGY - IMPROVED/MISSING

- BOUNDARY ELEMENTS FOR 3-D INELASTIC ANALYSIS
- BOUNDARY ELEMENTS FOR HOT FLUID/STRUCTURE INTERACTION
- EFFICIENT HYBRID ELEMENTS
- ADAPTIVE TRANSITIONAL FINITE ELEMENTS
- COMPUTATIONAL COMPOSITE MECHANICS
- COMPUTATIONAL CONTACT MECHANICS
- COUPLE COMPUTATIONAL SIMULATION WITH OPTIMIZATION

AN IMPORTANT PART OF THE CSM FOR ENGINE STRUCTURES PROGRAM IS THE IDENTIFICATION OF METHODOLOGY WHICH NEEDS IMPROVEMENT AND/OR IS MISSING. THIS METHODOLOGY INCLUDES SEVERAL KEY ELEMENTS AS LISTED IN THE ACCOMPANYING CHART.

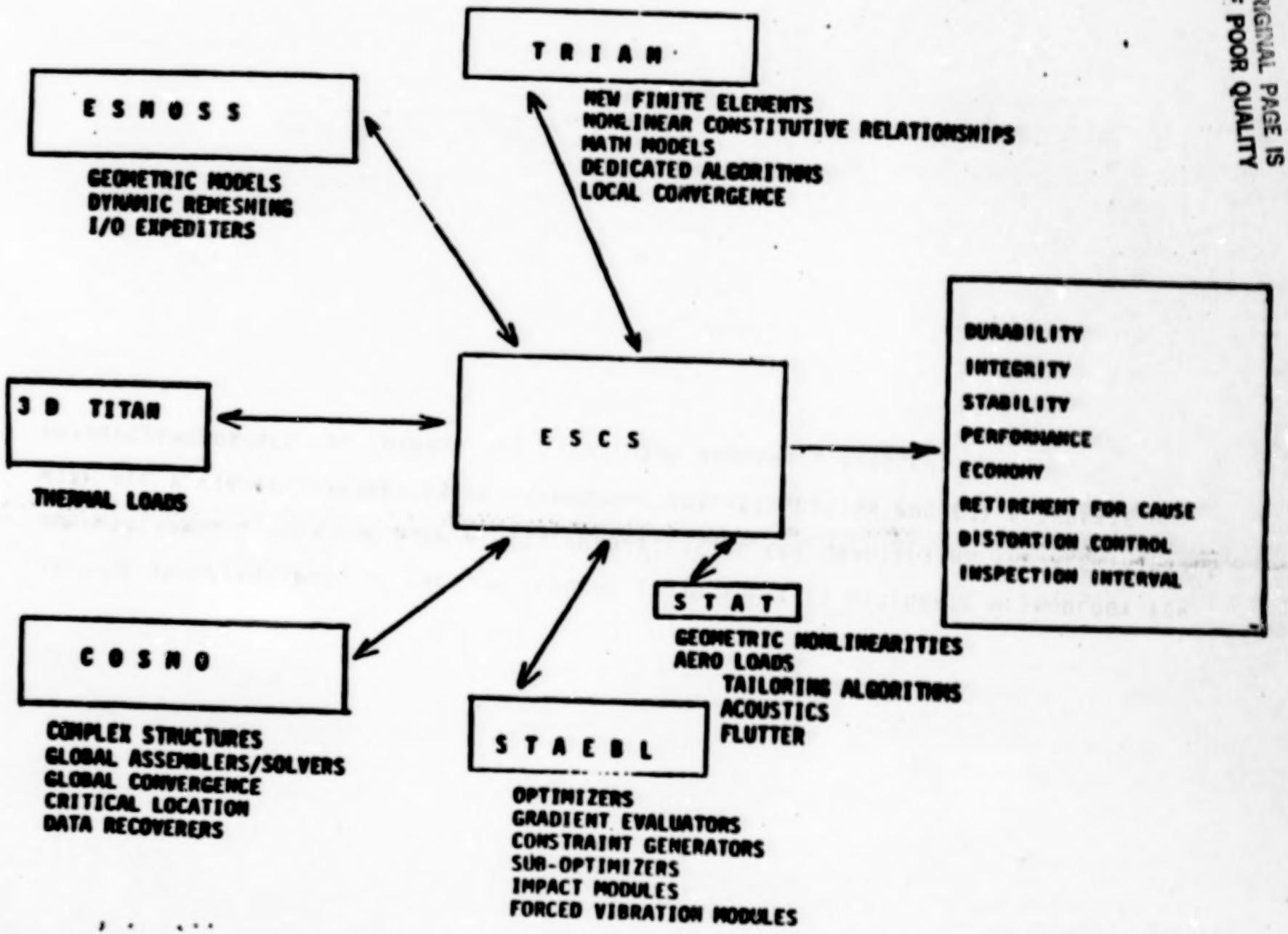
COMPUTATIONAL STRUCTURAL MECHANICS
IDENTIFIED METHODOLOGY - ALTERNATE

- PROBABILISTIC/STOCHASTIC:
  - VARIATIONAL PRINCIPLES FOR PROBABILISTIC FINITE ELEMENT
  - PROBABILISTIC STRUCTURAL ANALYSIS METHODS
  - PROBABILISTIC FRACTURE MECHANICS

- ALTERNATE FORMULATIONS:
  - MULTI-PARALLEL PROCESSORS FOR MULTI-DISCIPLINE MECHANICS PROBLEMS
  - SPECIALTY FUNCTIONS FOR SINGULAR MECHANICS PROBLEMS
  - COUPLED CONSTITUTIVE RELATIONSHIPS
  - DEDICATED EXPERT SYSTEMS

ANOTHER IMPORTANT PART OF THE CSM PROGRAM IS TO IDENTIFY ALTERNATE METHODLOGY FOR COMPUTATIONAL SIMULATION SUCH AS (1) PROBABILISTIC FOR QUANTIFYING THE ACERTAINTIES WITH ALL VARIABLES/PARAMETERS OF STRUCTURAL ANALYSIS/DESIGN AND (2) ALTERNATE METHODS/APPROACHES FOR FORMULATING STRUCTURAL MECHANICS PROBLEMS.

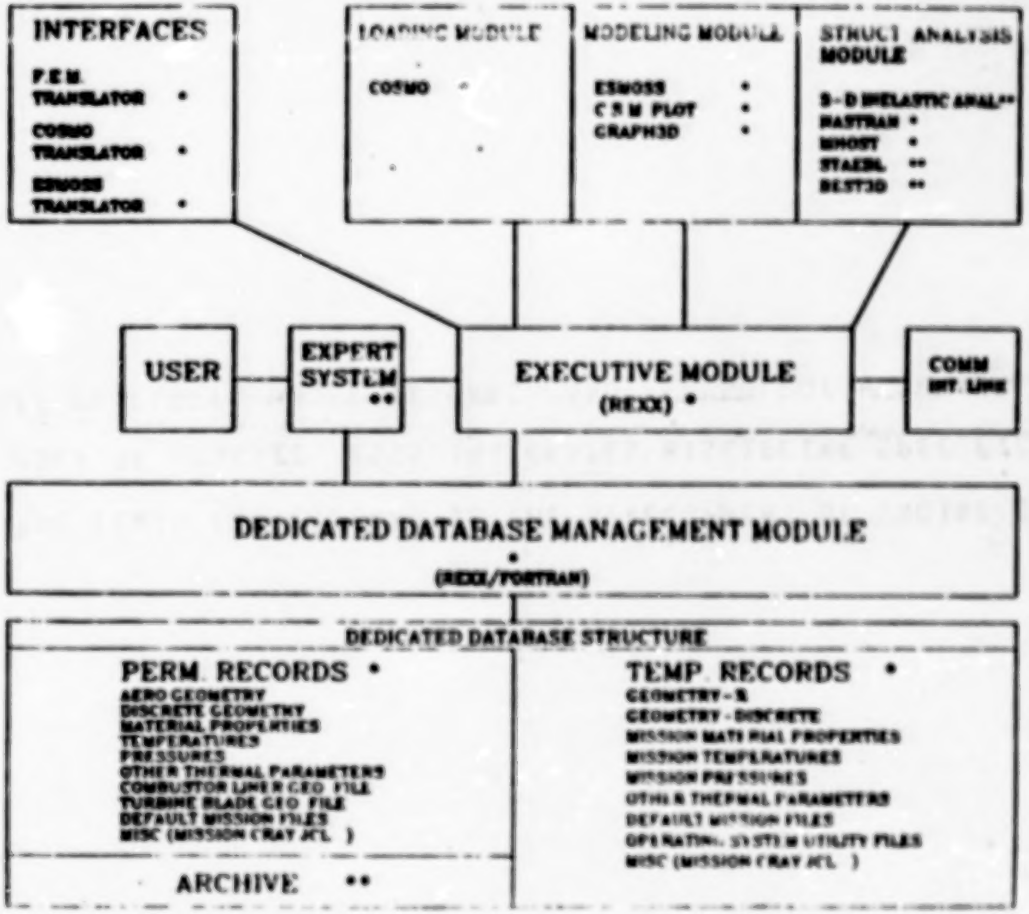
ENGINE STRUCTURES COMPUTATIONAL SIMULATOR (ESCS)



ORIGINAL PAGE IS
OF POOR QUALITY

A MAJOR PART OF THE LEWIS CSM PROGRAM IS THE DEVELOPMENT OF ENGINE STRUCTURES COMPUTATIONAL SIMULATOR (ESCS). ESCS INTEGRATES DISCIPLINE SPECIFIC METHODOLOGY AND COMPUTER CODES DEVELOPED UNDER RESEARCH AND TECHNOLOGY PROGRAMS.

SIMULATOR ARCHITECTURE OF THE SOFTWARE SYSTEM



• - PRELIMINARY VERSION AVAILABLE

** TO BE INSTALLED

ORIGINAL PAGE IS
OF POOR QUALITY

ESCS IS MODULAR WITH AN EXPERT SYSTEM DRIVEN EXECUTIVE MODULE. IT INCLUDES INTERFACING MODULES, A DATABASE AND ITS MANAGER. A SCHEMATIC OF THE ESCS PRESENT STATUS CONFIGURATION IS SHOWN IN THE ACCOMPANYING CHART.

ORIGINAL PAGE IS
OF POOR QUALITY

ENGINE STRUCTURES COMPUTATIONAL SIMULATOR

STRUCTURES DIVISION

STRUCTURAL MECHANICS BRANCH

ENGINE

FINITE ELEMENT MODEL

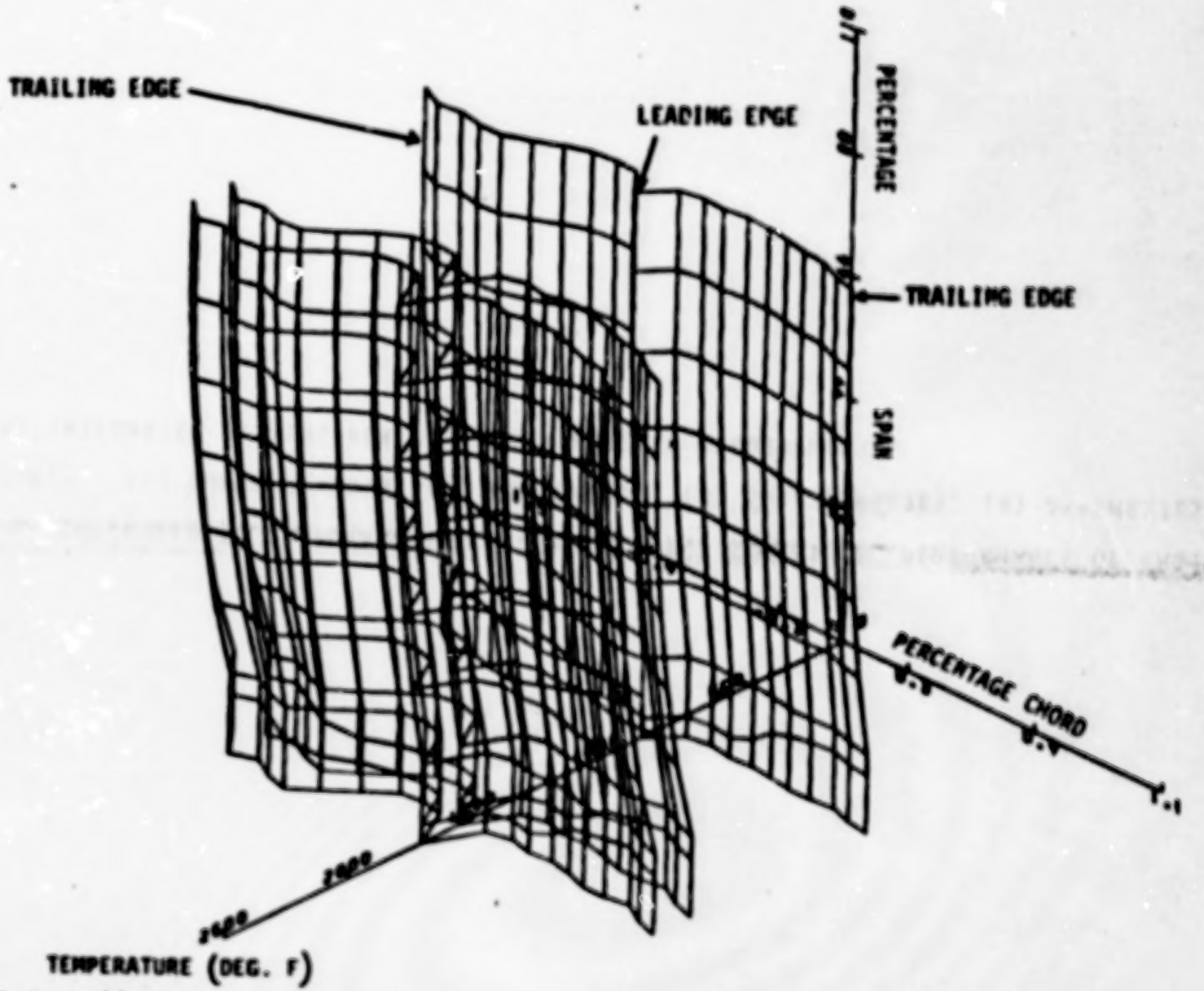
ROTOR STAGE

ROTOR SECTOR

BLADE

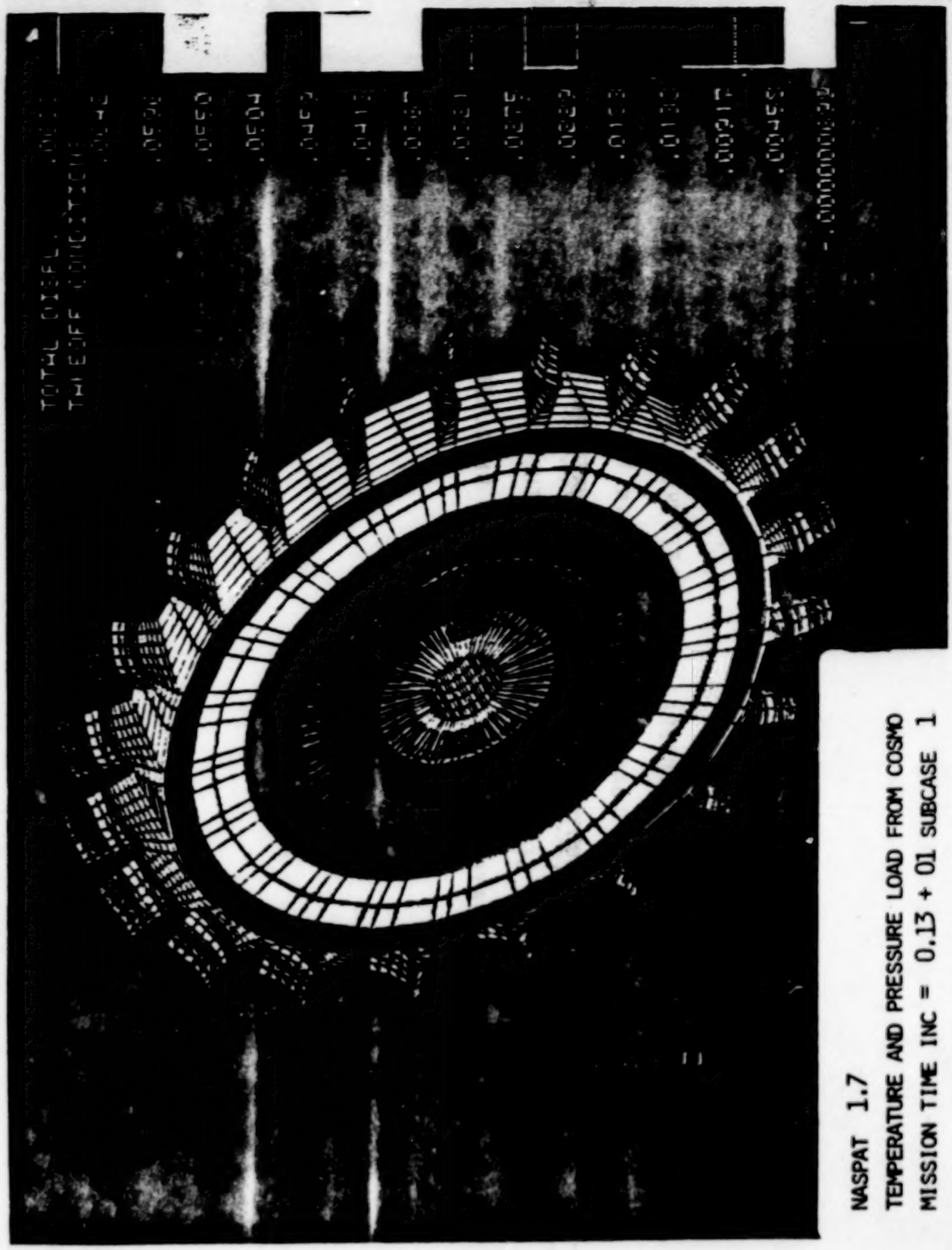
ESCS IS CONFIGURED TO COMPUTATIONALLY SIMULATE THE STRUCTURAL PERFORMANCE OF ENGINE STRUCTURES: (1) SUBCOMPONENTS, (2) COMPONENTS, (3) SUBASSEMBLIES, (4) ASSEMBLIES AND (5) INTEGRATED SYSTEMS FOR MISSION SPECIFIED REQUIREMENTS.

SURFACE TEMPERATURE PROFILE FOR TURBINE BLADE



THE LOADS ON THE BLADES (TEMPERATURES, PRESSURES AND ROTATING SPEEDS) ARE DETERMINED BY AN ENGINE LOADS MODULE (COSMO IN THE ESCS SCHEMATIC). THIS MODULE IS BASED ON ENGINE THERMODYNAMICS. THE TEMPERATURES AND PRESSURES ARE PREDICTED ON THE SURFACE AT USER SELECTED SPAN STATIONS. THE ACCOMPANYING CHART IS A TYPICAL EXAMPLE FOR TEMPERATURES. THE BLADE HAS BEEN UNFOLDED FOR 3-D PLOTTING PRESENTATION.

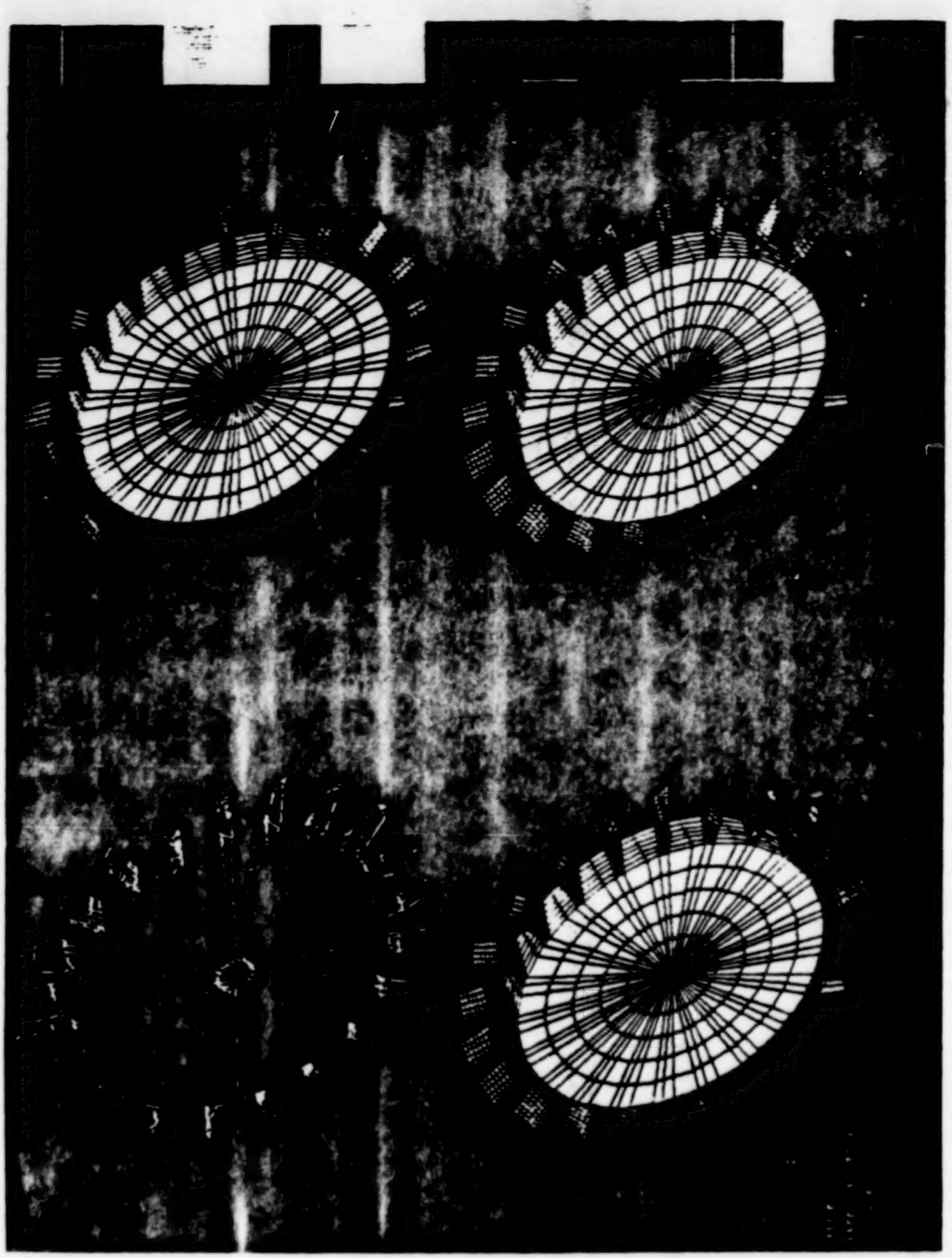
ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH



NASPAT 1.7
TEMPERATURE AND PRESSURE LOAD FROM COSMO
MISSION TIME INC = 0.13 + 01 SUBCASE 1

C-2

THE PRESSURE IS SIMILARLY REPRESENTED IN A 3-D PLOT.

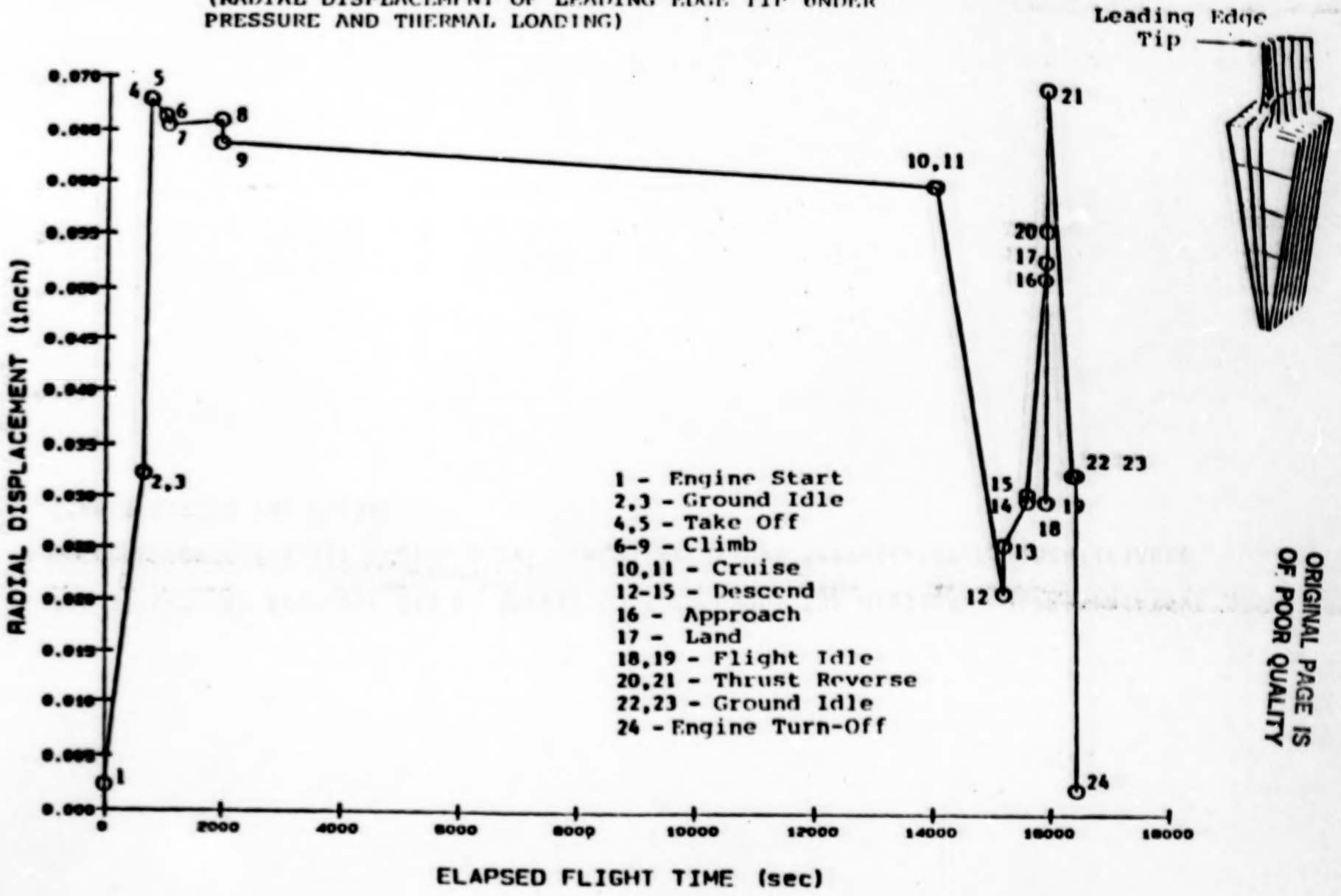


ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

THE STRUCTURAL RESPONSE CAN BE PREDICTED THROUGHOUT THE MISSION. REPRESENTATIVE RESULTS FOR BLADE-TIP RADIAL DISPLACEMENT ARE SHOWN GRAPHICALLY AT IDENTIFIABLE STAGES DURING THE FLIGHT.

MHOST AS A MODULE IN THE ENGINE STRUCTURES COMPUTATIONAL SIMULATOR (CSM)

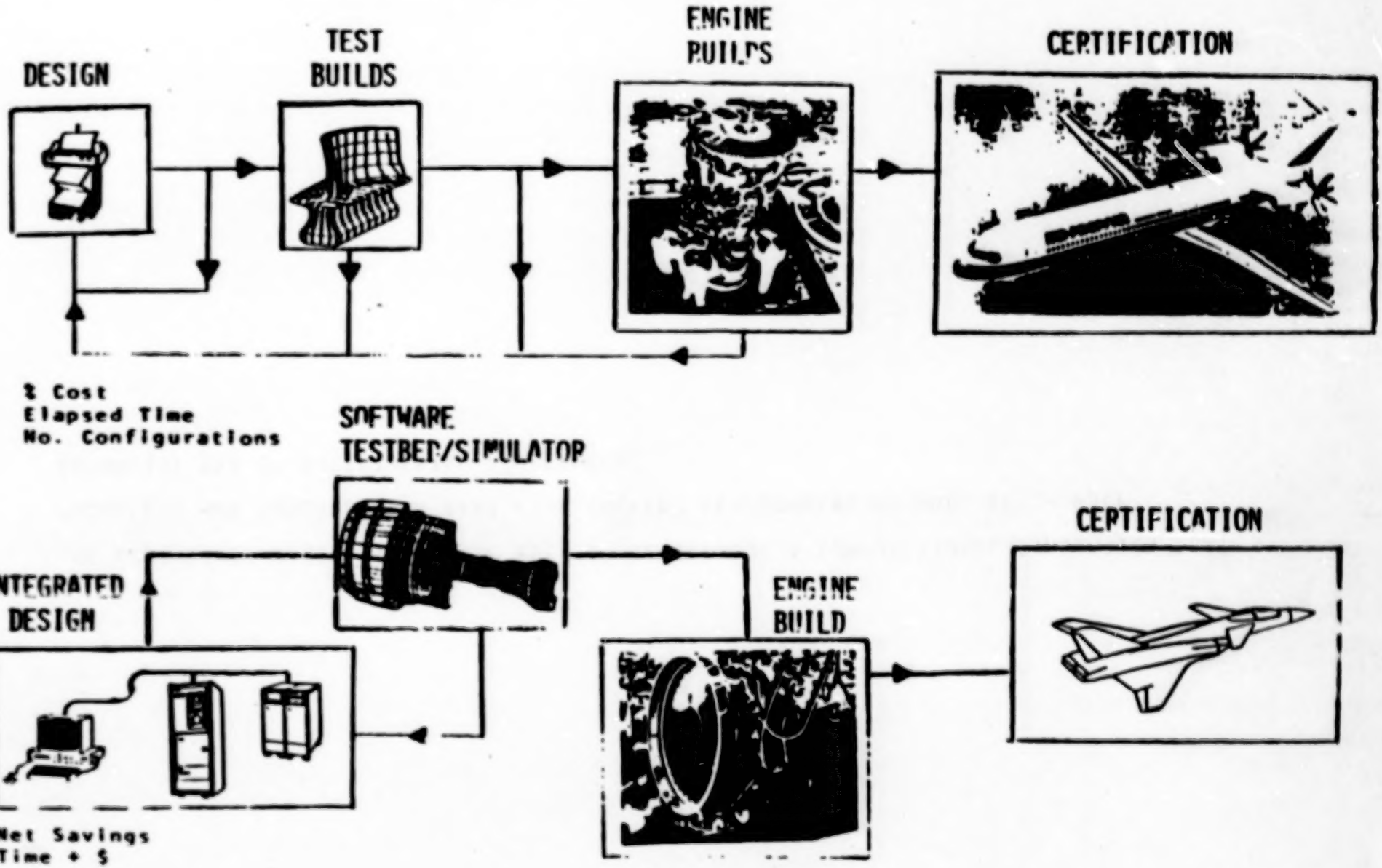
(RADIAL DISPLACEMENT OF LEADING EDGE TIP UNDER PRESSURE AND THERMAL LOADING)



ORIGINAL PAGE IS OF POOR QUALITY

THE LONG RANGE OBJECTIVE OF THE ESCS IS TO PROVIDE A COMPUTATIONAL SIMULATION THAT PARALLELS AND REPLACES, IN PART, THE CURRENT DEVELOPMENT METHODS WHICH MAKE EXTENSIVE USE OF EXPERIMENTAL PROCEDURES.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH



PRECEDING PAGE BLANK NOT FILMED

PAGE 483 INTENTIONALLY BLANK

POTENTIAL BENEFITS TO AEROSPACE INDUSTRY

- REDUCED DEVELOPMENT TIME AND COSTS
- FEWER DEVELOPMENT ENGINE BUILDS
- LONGER LIFE COMPONENTS
- REDUCED LIFE CYCLE COSTS ON COMPONENTS
- REDUCED COMPONENT AND ENGINE WEIGHT
- IMPROVED ENGINEERING PRODUCTIVITY
- INCREASED PERFORMANCE

THE ANTICIPATED BENEFITS OF ESCS ARE SUMMARIZED, QUALITATIVELY, IN THE LAST CHART.

N89
29793
UNCLAS

N89-29793

54-39
298
211961
S6006000

INTERFACING MODULES FOR INTEGRATING DISCIPLINE SPECIFIC STRUCTURAL MECHANICS CODES

proof — correct title

by

Ned M. Endres
Sverdrup Tech Inc.
NASA LeRC Group

The purpose ~~of this presentation~~ is to outline the organization and capabilities of the Engine Structures Computational Simulator (Simulator) at NASA Lewis Research Center. One of the goals of ~~our~~ the research at Lewis is to integrate our various discipline specific structural mechanics codes into a software system which can be brought to bear effectively on a wide range of engineering problems. This system must possess the qualities of being effective and efficient while still remaining "user friendly". The simulator was initially designed for the finite element simulation of gas jet engine components. Currently, the simulator has been restricted to only the analysis of high pressure turbine blades and the accompanying rotor assembly, although the current installation can be expanded for other applications. The simulator presently assists the user throughout its procedures by performing information management tasks, executing external support tasks, organizing analysis modules and executing these modules in the user defined order while maintaining processing continuity.

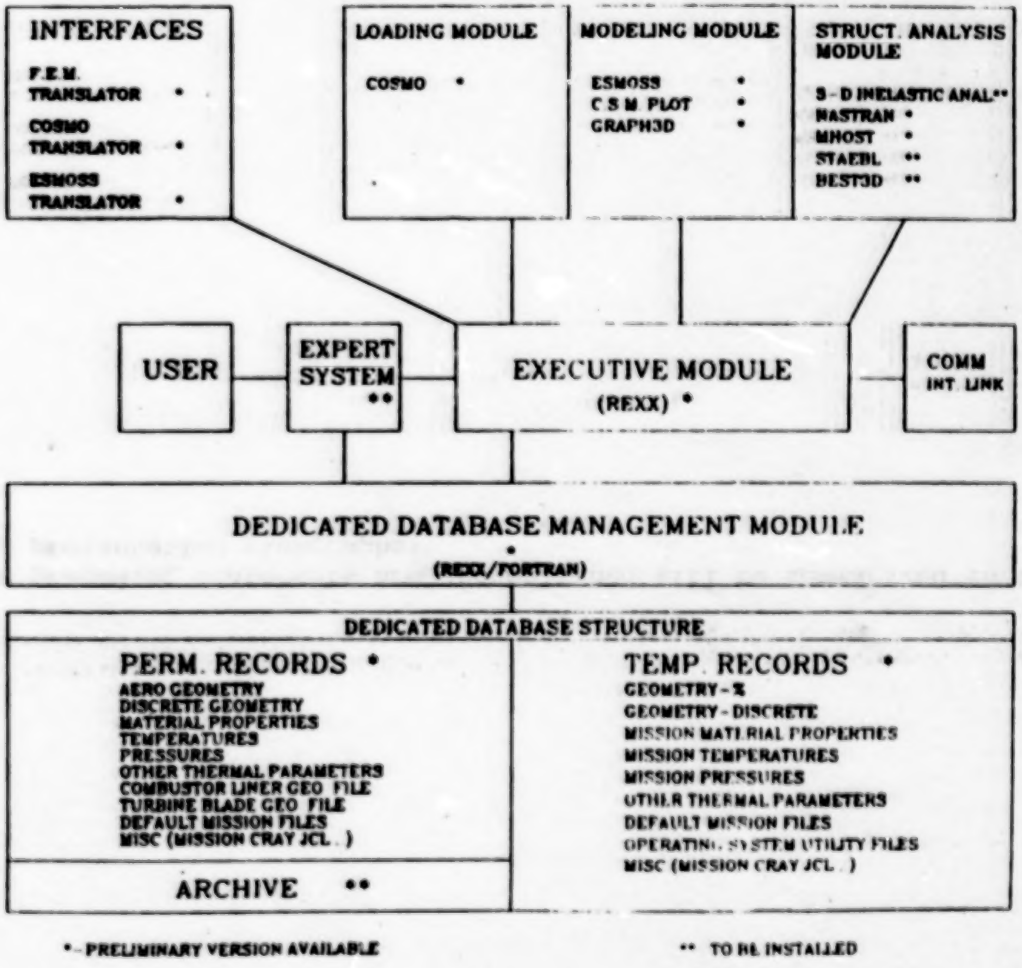
PRECEDING PAGE BLANK NOT FILMED

PAGE 486 INTENTIONALLY BLANK

HOW?

How the preceding statements are accomplished will be summarized in the following presentation viewgraphs:

SIMULATOR ARCHITECTURE OF THE SOFTWARE SYSTEM



• - PRELIMINARY VERSION AVAILABLE

** TO BE INSTALLED

ORIGINAL PAGE IS
OF POOR QUALITY

SIMULATOR – EXPERT SYSTEM

CAPABILITIES/FEATURES:

1). FORMULATION OF PROBLEM USING ACQUIRED AND USER KNOWLEDGE.
2). PROVIDE EXPERT OPINION AND ASSISTANCE IN FORMULATION PROCESS.
3). CONSTRUCT EXECUTIVE MODULE COMMANDS.
4). ACQUIRE AND MAINTAIN KNOWLEDGE BASE.
5). FORMULATE INFORMATION MANAGEMENT SYSTEM REQUESTS.

SUBMODULES:

1). RULE BASE.
2). KNOWLEDGE BASE.
3). INFERENCE ENGINE.

PRECEDING PAGE BLANK NOT FILMED

INTENTIONALLY BLANK

The simulator can be decomposed into three (3) distinct collections of modules; the Expert System / Executive modules, the Interface / Processing Codes and the Dedicated Database Management module / Dedicated Database. The Expert System and Executive modules dictate and maintain processing flow continuity. The Interface modules and Processing codes are the vehicle for the performance of the analysis. While the Dedicated Database Management module and Database provide and maintain the information required for the analysis.

SIMULATOR–EXECUTIVE MODULE

SETUP FACILITIES

MAIN SUBMODULE

(INTERACTIVE CAPABILITY)

(REXX)

SUBMODULES

CRAY STATION MODULE	DEFAULT STORAGE RETRIEVAL
CRAY MONITOR MODULE	COSMO PREPROCESSOR JCL
CRAY STANDBY MODULE	COSMO MISSION MODULE JCL
CRAY JCL BUILD MODULE	COSMO COMB. MODULE JCL
EXEC. BACKUP MODULE	COSMO VANE MODULE JCL
F.E.M. ANAL. ASSEMBLY MOD.	COSMO BLADE MODULE JCL
VIRTUAL MACHINE SCANNER	COSMO POSTPROCESSOR JCL
TEMP. STORAGE SETUP	NASTRAN PREPROCESSOR JCL
STORAGE MONITOR	MHOST PREPROCESSOR JCL
VIRT. MACH. RDR SCANNER	ESMOSS PREPROCESSOR JCL
EXEC. BAILOUT MODULE	ESMOSS POSTPROCESSOR JCL
BACKGROUND MODULE	EXEC. HELP MODULE
BUFFER CHECK MODULE	HELP PLOTTING MODULE
INTRODUCTION MODULE	MENU MODULE

PAGE 495 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

SIMULATOR – EXECUTIVE MODULE

CAPABILITIES/FEATURES:

1). CONTROL PROCESSING FLOW.
2). MONITOR/INTERPRET DIRECTIVE DIALOG FROM EXPERT SYSTEM.
3). ASSEMBLE AND MODIFY F.E. ANALYSIS INFORMATION.
4). MANAGE INFORMATION FLOW BETWEEN MODULES.
5). ITERATIVE PROCESSING.
6). AUTOMATED FILE MANAGEMENT.
7). AUTOMATED INFORMATION RETRIEVAL SYSTEM.
8). AUTOMATED TEMPORARY STORAGE ALLOCATION.
9). MENU SELECTION OF PROCESSING FLOW.
10). TWO (2) F.E. CODES ARE CURRENTLY AVAILABLE: NASTRAN MHOST.
11). ONE (1) THERMODYNAMIC MODELLING CODE IS AVAILABLE: COSMO.
12). DEFAULT PROCESSING FLOW AVAILABLE.
13). CRAY FACILITY.
14). DEFAULT SAMPLE CASE AVAILABLE.
15). INTERACTIVE OR AUTOMATED ASSEMBLY OF F.E. ANALYSIS INPUT.
16). AUTOMATED TEMPORARY DISK USAGE MONITOR.
17). USER SELECTABLE MISSION PROFILE.
18). USER SELECTABLE MISSION TIME INCREMENTS.
19). USER SELECTABLE GEOMETRY.
20). TWO DEFAULT GEOMETRIES PROVIDED.

SUBMODULES:

1). DEFAULT STORAGE RETRIEVAL.
2). RESIDENT MECHANICS CODES JCL.
3). CRAY FACILITY.
4). F.E.M. ANALYSIS ASSEMBLY.
5). SETUP.

The primary function of the Executive module is to control the flow of the data processing within the simulator. The Executive module is composed of twenty eight (28) submodules. Each submodule performs a specific group of tasks (code execution, database data retrieval, storage monitoring, etc...) The execution of these tasks within the Executive module is controlled by interpreted Expert system commands or interactive user participation. These commands in turn lead to the execution of the Executive submodule tasks. Collectively, these submodules perform the tasks needed for the completion of the finite element analysis.

SIMULATOR–DEDICATED DATABASE MANAGEMENT

MAIN SUBMODULE (REXX) (INTERACTIVE/DIRECT CAPABILITY)
INTERACTIVE DATA RETRIEVAL (REXX/FORTRAN 77)
INDEXES
PERMANENT RECORDS INDEXES (RESIDENT RECORDS INDEXES)
TEMPORARY RECORDS INDEXES (GENERATED RECORDS INDEXES)
DATA STORAGE PARAMETER FACILITY (COMMAND LEVEL)

PAGE 499 INTENTIONALLY BLANK

SIMULATOR – DEDICATED DATABASE MANAGEMENT

CAPABILITIES/FEATURES:

1). RETRIEVE AND UPDATE INFORMATION FROM PERMANENT OR TEMPORARY RECORDS VIA GENERIC INFORMATION REQUESTS.
2). RECEIVES GENERIC INFORMATIONAL REQUESTS FROM THE USER OR EXECUTIVE MODULE.
3). USES LIST DIRECTED SEARCH FOR LOCATING INFORMATION.
4). USER TRANSPARENT STORAGE SELECTION.
5). NAME ONLY REQUESTS FOR INFORMATION RETRIEVAL.
6). AUTOMATED LIST MANAGEMENT.

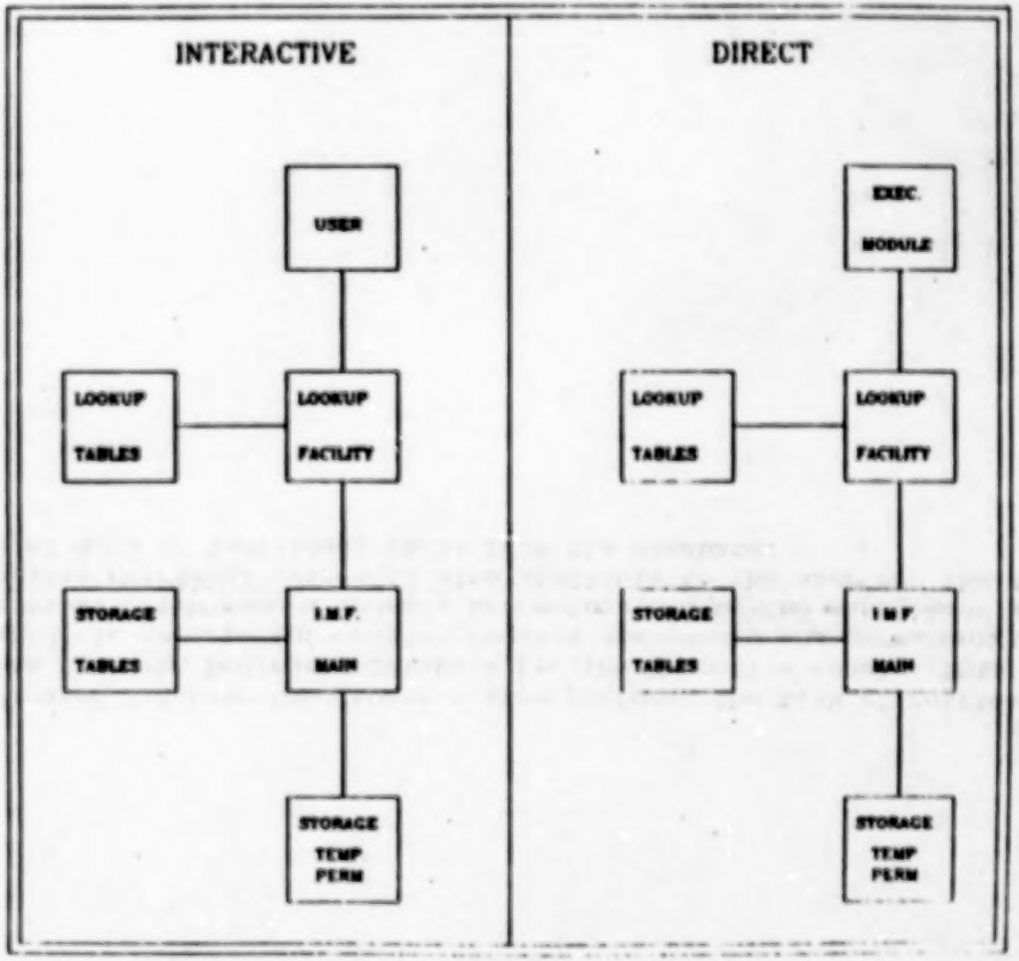
SUBMODULES:

1). MAIN SUBMODULE (REXX).
2). INTERACTIVE DATA RETRIEVAL (REXX/FORTRAN 77).
3). RECORD AND LOOKUP INDEXES.

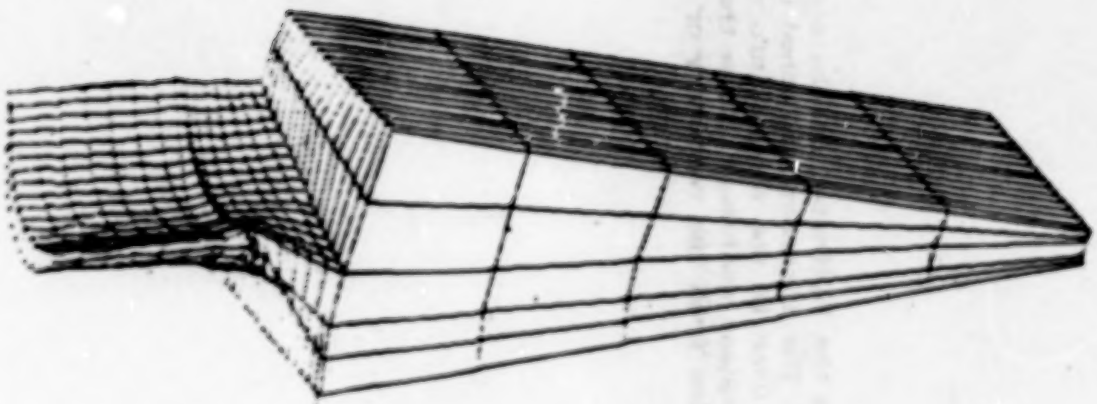
PRECEDING PAGE BLANK NOT FILMED

The Dedicated Database Management system performs the task of retrieving information from the Dedicated Database for the Executive module. This is accomplished via generic information request statements and permanent/temporary record indexes. These record indexes are maintained by the management system. An interactive retrieval system is also available to the user for the extraction of blocks of data or individual facts from the database.

INTERACTIVE DATA RETRIEVAL FACILITY



The information from the Dedicated Database can be retrieved in two methods; interactive or direct. The interactive method utilizes lookup indexes and tables in order to guide the user to the needed information. The direct method involves generic information request statements from the Executive module which directly retrieve the information from the temporary or permanent database.



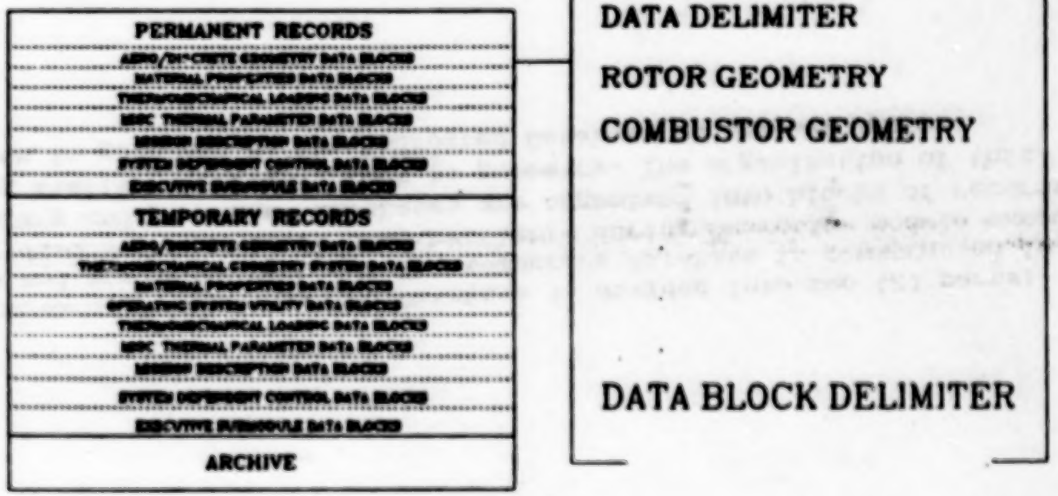
This is a plot illustrating the type of information which resides within the permanent database. This figure represents a sector of high pressure turbine blade rotor assembly, a small portion of a single stage of the jet engine. This geometry and accompanying flow information can reside within the database under one generic name, in this case known as SECTOR.

SIMULATOR–DEDICATED DATABASE STRUCTURE

PERMANENT RECORDS
AERO/DISCRETE GEOMETRY DATA BLOCKS
MATERIAL PROPERTIES DATA BLOCKS
THERMOMECHANICAL LOADING DATA BLOCKS
MISC. THERMAL PARAMETER DATA BLOCKS
MISSION DESCRIPTION DATA BLOCKS
SYSTEM DEPENDENT CONTROL DATA BLOCKS
EXECUTIVE SUBMODULE DATA BLOCKS
TEMPORARY RECORDS
AERO/DISCRETE GEOMETRY DATA BLOCKS
THERMOMECHANICAL GEOMETRY SYSTEM DATA BLOCKS
MATERIAL PROPERTIES DATA BLOCKS
OPERATING SYSTEM UTILITY DATA BLOCKS
THERMOMECHANICAL LOADING DATA BLOCKS
MISC. THERMAL PARAMETER DATA BLOCKS
MISSION DESCRIPTION DATA BLOCKS
SYSTEM DEPENDENT CONTROL DATA BLOCKS
EXECUTIVE SUBMODULE DATA BLOCKS
ARCHIVE

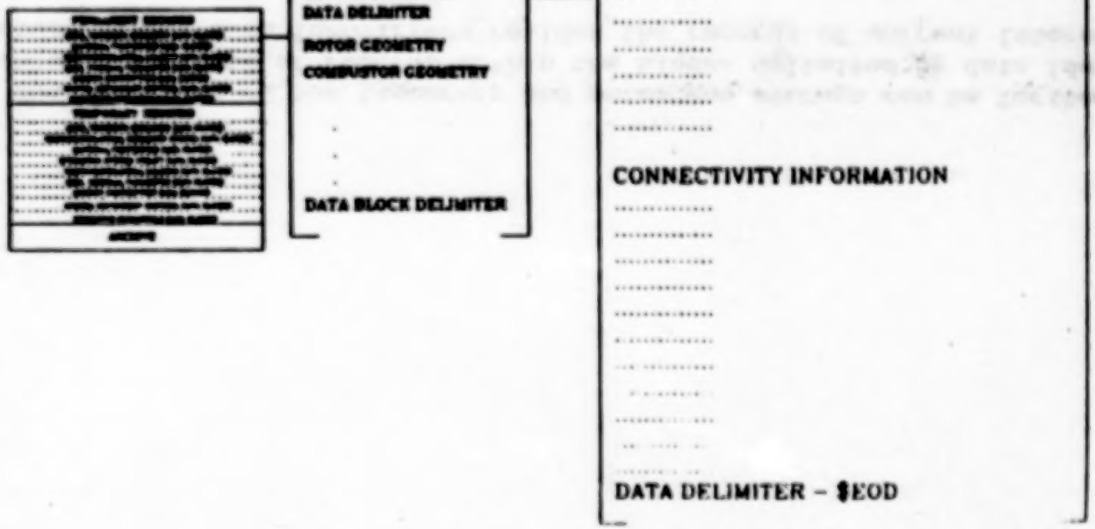
The structure of the Dedicated Database is divided into two (2) parts; permanent and temporary records. The temporary database is constructed in temporary disc storage and is only available during Executive module execution. The temporary and permanent databases are organized into blocks of records all possessing similar qualities, such as geometry. The organization of this information is governed by the Dedicated Database management system.

SIMULATOR–DEDICATED DATABASE STRUCTURE



The structure of the temporary and permanent storage can be further subdivided groups of records within the blocks delimited by data identifiers. Between these data identifiers resides the records of current interest.

SIMULATOR–DEDICATED DATABASE STRUCTURE



ORIGINAL PAGE IS
OF POOR QUALITY

SIMULATOR – DEDICATED DATABASE STRUCTURE

CAPABILITIES/FEARTURES:

1). A TEMPORARY OR PERMANENT REPOSITORY TO MAINTAIN INFORMATION FOR USE BY EXECUTIVE AND EXPERT SYSTEM MODULES.
2). ITEMS OF INFORMATION ARE STRICTLY ORGANIZED AND THEIR LOCATIONS ARE DOCUMENTED.

SUBMODULES:

1). PERMANENT RECORDS.
2). PERMANENT RECORD INDEXES.
3). TEMPORARY RECORDS.
4). TEMPORARY RECORD INDEXES.

PRECEDING PAGE BLANK NOT FILMED

The data identifier is expressed as a simple mnemonic characterizing the contents of that group of data. This mnemonic is not used by the database management system for retrieval, but it can be used by the user to identify the contents of the database directly. The Dedicated Database Management system maintains its own set of indexes for identifying and locating information independent of the data itself. So, within the Dedicated Database there are a collection of data blocks which each contain data identifiers and data. Collectively, this information is available, and necessary, for the execution of the Executive module.

SIMULATOR INTERFACE MODULES

COSMO INTERFACE MODULES

COSMO PREPROCESSOR MODULE

GENERIC F.E.M. INFORMATION ----------> COSMO COMPATIBLE F.E.M. INFORMATION

COSMO POSTPROCESSOR MODULE

COSMO COMPATIBLE F.E.M. INFORMATION ----------> GENERIC F.E.M. INFORMATION

NASTRAN INTERFACE MODULES

GENERIC F.E.M. INFORMATION ----------> NASTRAN INPUT F.E.M. INFORMATION

MHOST INTERFACE MODULES

GENERIC F.E.M. INFORMATION ----------> MHOST INPUT F.E.M. INFORMATION

ESMOSS INTERFACE MODULES

ESMOSS PREPROCESSOR MODULE

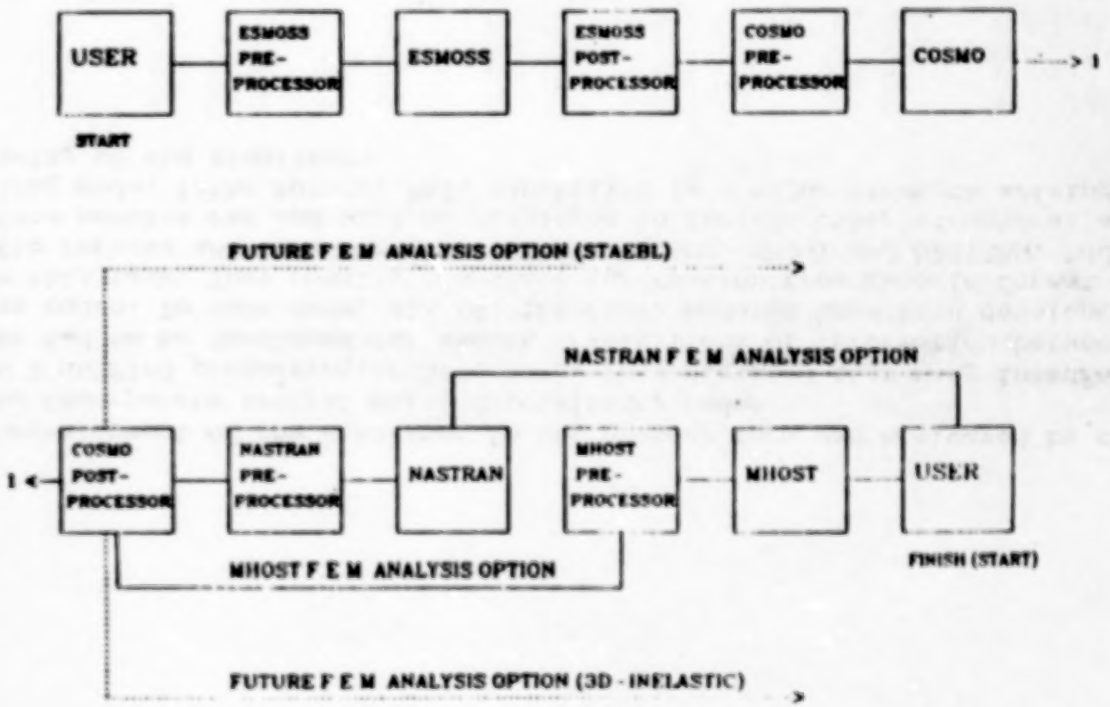
ESMOSS GEOMETRY MODELLING COMMANDS ----------> O.S. BUFFER COMMANDS

ESMOSS POSTPROCESSOR MODULE

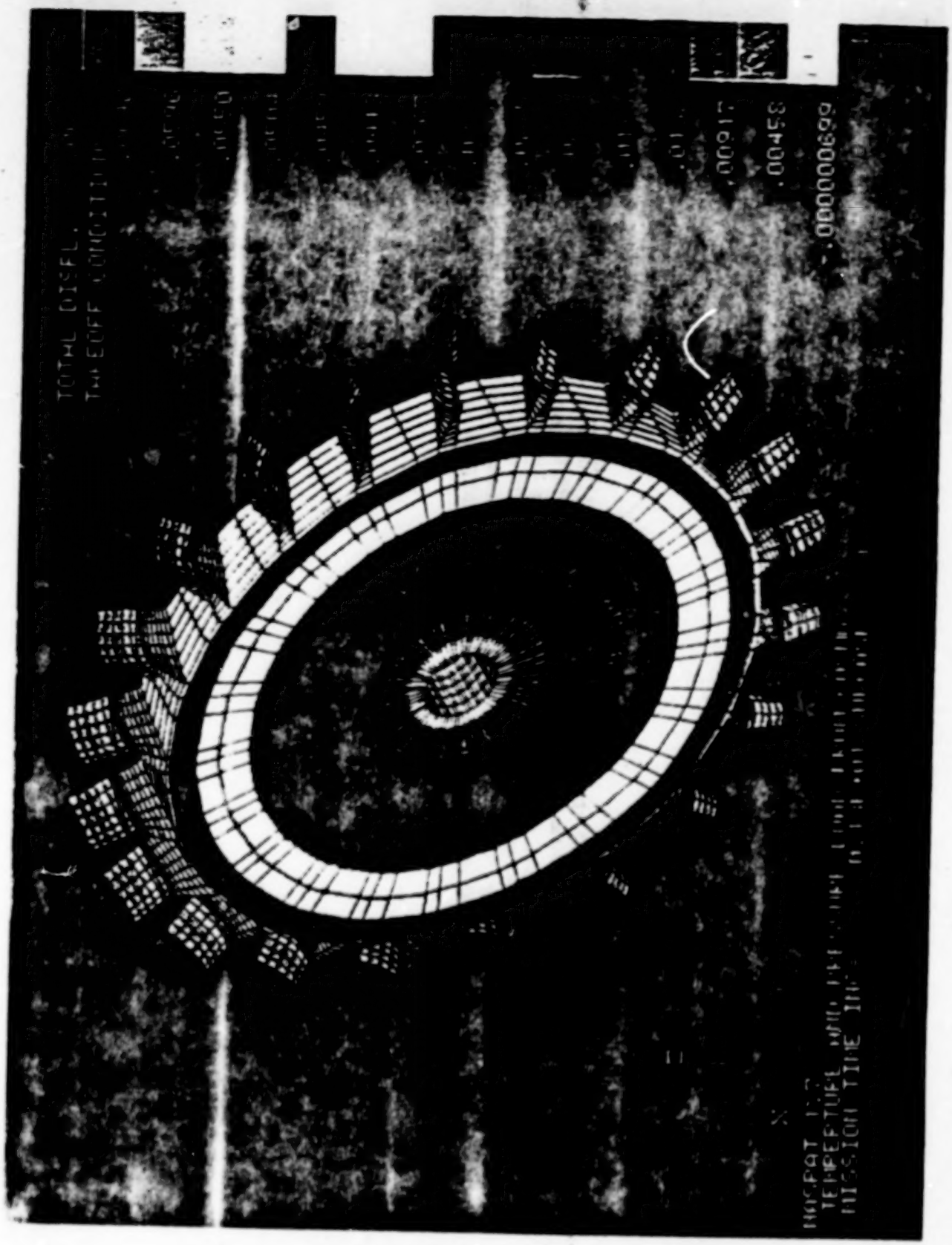
ESMOSS GEOMETRY F.E.M. INFORMATION ----------> GENERIC F.E.M. INFORMATION

The development of the simulator in its present form was motivated by the need to incorporate several multi-disciplinary codes within a unified processing architecture. This dictated that many interface modules had to be developed for smooth tranisitions of information between the various codes. To this date, six (6) interface modules have been developed to handle this task. They readily translate information from generic format to code specific formats and visa versa for ESMOSS, COSMO, MHOST and NASTRAN. Additional interface modules can and will be developed to include other structural and modelling codes inthe future. This capability is a major strength arising from modularity of the simulator.

SIMULATOR PROCESSING FLOW LOGIC



A sample of the processing flow logic is illustrated in the slide on the opposite page. In this example, the user formulates the finite element model with the ESMOSS preprocessor, ESMOSS and the ESMOSS postprocessor combination. Next, the thermodynamic loading for the flight is developed with the COSMO preprocessor, COSMO, and the COSMO postprocessor combination. Finally, the finite element analysis is conducted with NASTRAN or MHOST.



ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

A sample of the type of analysis results that can be achieved are shown on the opposite page. In this case, an entire rotor-blade assembly was simulated throughout the flight with only the takeoff conditions being shown here. The slide displays the magnitude of the total displacement vector for the case of temperature, pressure and rotational loadings at takeoff condition under quasi-static equilibrium conditions. The finite analysis, in this case, was conducted with NASTRAN.

N89
2979
UNCLAS

CSM RESEARCH: METHODS AND APPLICATION STUDIES

Norman F. Knight, Jr.†
NASA Langley Research Center

INTRODUCTION

Computational mechanics is that discipline of applied science and engineering devoted to the study of physical phenomena by means of computational methods based on mathematical modeling and simulation, utilizing digital computers.[1] The discipline combines theoretical and applied mechanics, approximation theory, numerical analysis, and computer science. Computational mechanics has had a major impact on engineering analysis and design. When applied to structural mechanics, the discipline is referred to herein as computational structural mechanics.

Complex structures being considered by the NASA for the 1990's include composite primary aircraft structures and the space station. These structures will be much more difficult to analyze than today's structures and necessitate a major upgrade in computerized structural analysis technology. NASA has initiated a research activity in structural analysis called Computational Structural Mechanics or CSM. The broad objective of the CSM activity is to develop advanced structural analysis technology that will exploit modern and emerging computers — such as computers with vector and/or parallel processing capabilities.

The Langley CSM activity, initiated in October 1984, is described in reference 2. The present paper describes the current research directions for the Methods and Application Studies Team of the Langley CSM activity.

† Aerospace Engineer, Structural Mechanics Branch, Structures and Dynamics Division.

N89-29794

PRECEDING PAGE BLANK NOT FILMED

PAGE 520 INTENTIONALLY BLANK

LANGLEY CSM PROGRAM

Parallel Structural Methods Research - Olaf O. Storaasli

Testbed Development - Ronnie Gillian

Methods and Applications Studies - Norm Knight

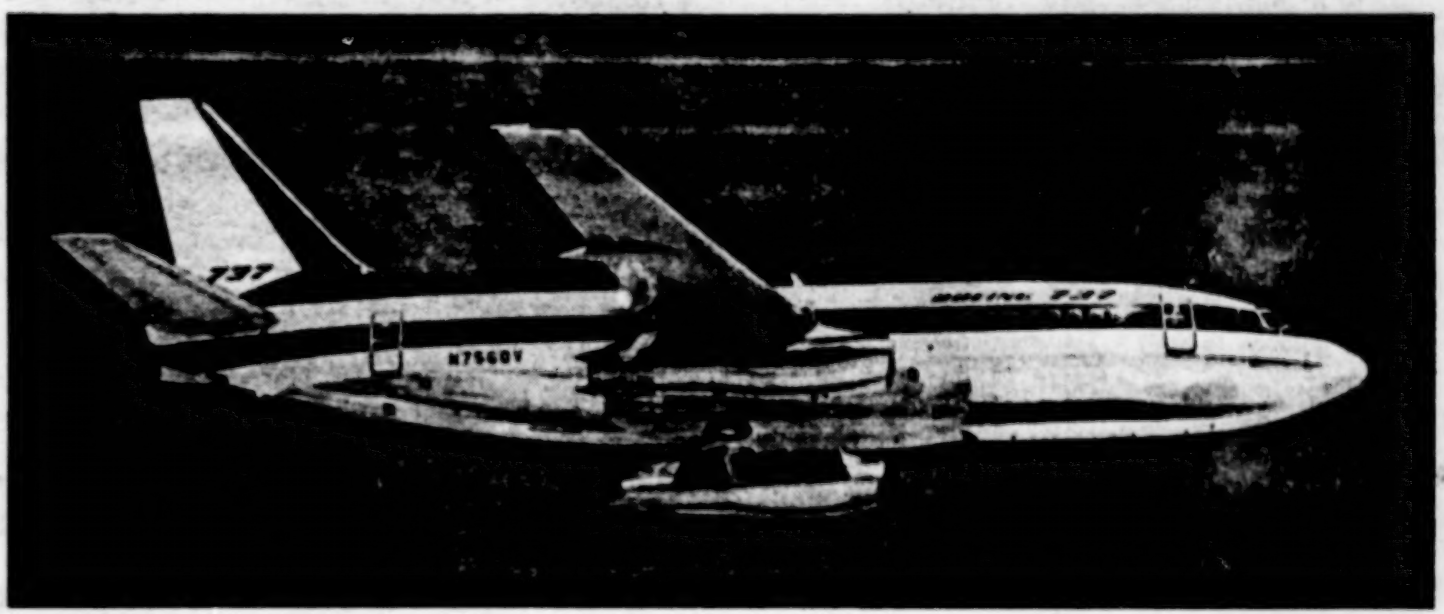
LANGLEY CSM PROGRAM

The Langley CSM program is organized as three teams. The first team's primary responsibility is parallel structural methods research. The second team's primary responsibility is testbed development. The third team's primary responsibility is methods and application studies. Each team interacts with the other teams to achieve a synergistic effect. This paper will describe the objective and research directions of the third team.

The third team's primary responsibilities are to develop structural analysis methods and carry out application studies. The third team emphasizes advanced structural analysis methods — such as 3-D stress analysis, composite laminate analysis, and error detection and control — that have application to all computers rather than focusing on methods that exploit MIMD computers, which is the responsibility of the parallel processing team.

STRUCTURAL ANALYSIS PROBLEMS

Commercial/military transport aircraft



Problem areas

- Composites
- Discontinuities
- Windows, doors, holes, damage
- Buckled skin
- Routine detailed analysis

Accurate and reliable analytical tools will lead to:

- Reduced testing needs
- Reduced weight associated with high margins for uncertainty
- Improved performance

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

STRUCTURAL ANALYSIS PROBLEMS
– Commercial/Military Transport Aircraft–

Structural analysis problems for commercial/military transport aircraft are indicated on this slide. These problems occur in metallic as well as composite structures. However, the brittle nature of composites requires that their strength limits and failure characteristics be well understood before composite structural components can be designed properly. In many respects, a greater need for reliable and accurate analytical predictive techniques exists for composite structures than for metallic structures. The difficulties associated with analyzing composite structures are magnified when there are discontinuities such as free edges, bolt holes, and bonded joints. Other problems are caused by windows, doors, access holes, and damage.

To save weight, many designers are proportioning structural panels so that the skin can buckle in service. Such panels are lighter than buckling-resistant panels. However, analyzing the postbuckling response of these structurally efficient panels is very difficult with today's analysis procedures. Specifically, it can be computationally difficult to track the postbuckling response; it can be frustrating and time consuming to have numerous restarts; and it can be expensive.

The problem of calculating detailed stress distributions around discontinuities in buckled, composite structural components for use with the various analytical failure prediction techniques has not been thoroughly explored. Because of the complex failure modes of composite structures, it may be necessary to perform a detailed 3-D stress analysis in a local region in order to obtain an adequate estimate of the strength. Today, carrying out such an analysis of a composite component can be a major research task. The capability to carry out, on a routine basis, a local 3-D stress analysis of a composite component within a larger 2-D analysis model is needed.

Accurate and reliable structural analysis procedures will lead to reduced time and cost for testing, reduced weight penalty associated with high margins for uncertainty, and improved performance. These technology improvements will be incorporated in structural analysis software, and that software will account for advancements in computer hardware. An appropriate match of structural analysis software and computer hardware could provide a substantial increase in computational speed.

METHODS AND APPLICATION STUDIES

OBJECTIVE: TO IDENTIFY, DEVELOP, AND EXTEND STRUCTURAL ANALYSIS AND COMPUTATIONAL METHODS THAT HAVE HIGH POTENTIAL

APPROACH: METHODS DEVELOPMENT DRIVEN BY STRUCTURAL APPLICATIONS AND ANALYSIS DEFICIENCIES

METHODS AND APPLICATION STUDIES

The objective of the methods and application studies team is to identify, develop, and extend structural analysis and computational methods that have high potential for solving critical application problems and for removing analysis deficiencies. Structural applications and analysis deficiencies drive the methods development and the application of these methods should provide new insight for complex, nonlinear structural mechanics problems. The analysis and computational methods should be amenable to error analysis. That is, given a physical problem and a mathematical model of that problem, an analyst would like to know the probable error in predicting a given response quantity. The ultimate goal is to specify the error tolerance and to use a self-adaptive procedure to adjust the mathematical model or solution strategy to obtain that accuracy.

CSM FOCUS PROBLEMS

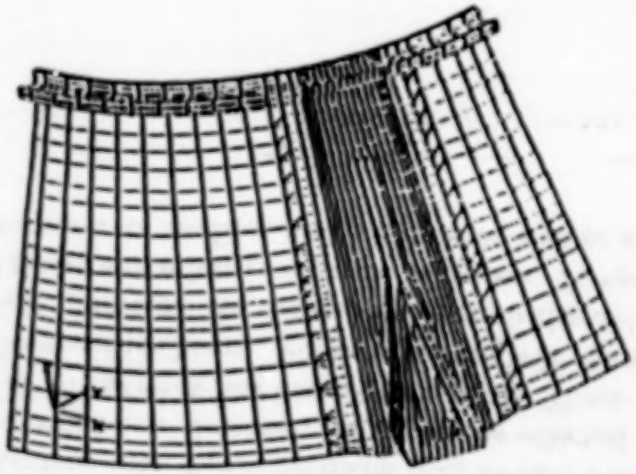
- CHALLENGING STRUCTURAL MECHANICS PROBLEMS THAT STRETCH OUR ABILITY TO PREDICT STRUCTURAL RESPONSE
- CHALLENGING COMPUTATIONAL PROBLEMS THAT STRETCH OUR COMPUTING LIMITS
- COMMON PROBLEMS FOR ALL CSM PARTICIPANTS LEADING TOWARDS FOCUSED EFFORT
- GUIDE METHODS RESEARCH AND DEVELOPMENT FOR GENERIC CLASS OF PROBLEMS
- COORDINATION WITH OTHER GROUP(S) NEEDED TO AUGMENT MANPOWER REQUIRED TO ANALYZE LARGE COMPLEX STRUCTURES

CSM FOCUS PROBLEMS

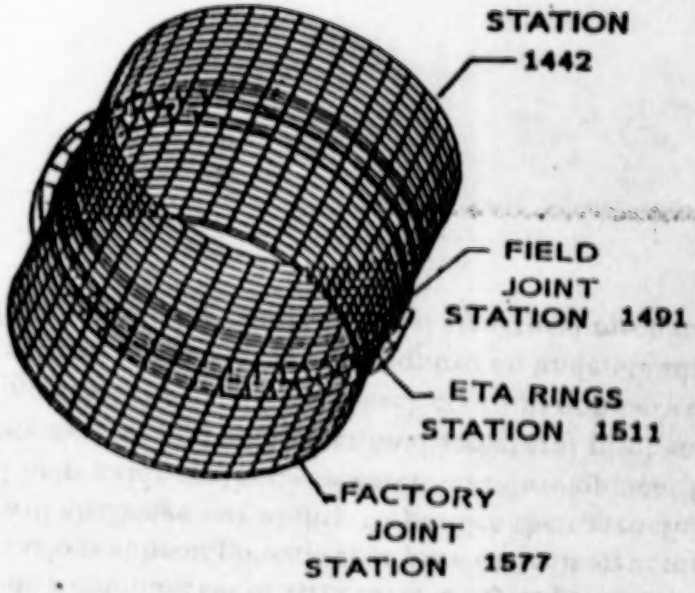
The Langley CSM activity employs the concept of focus problems to provide a common set of structural analysis problems for all CSM participants. Focus problems may be entire aerospace vehicles or various subcomponents that pose difficult structural mechanics problems. However, the problems selected as focus problems will challenge our ability to predict their structural response or will stretch our computing limits. These focus problems will help guide methods research and development for generic classes of problems. Focus problems will change as new technology evolves and computational structural mechanics methodology develops. As the size and complexity of the focus problems increase, the need for research coordination between the CSM group and other groups also increases. To use large, complex structures as focus problems requires an understanding of the structure, its loading, and life cycle as well as an understanding of the underlying computational structural mechanics issues.

SPACE-ORIENTED APPLICATION STUDIES

SOLID ROCKET BOOSTER (SRB)



SRB AFT SKIRT



SRB/ETA RING INTERFACE

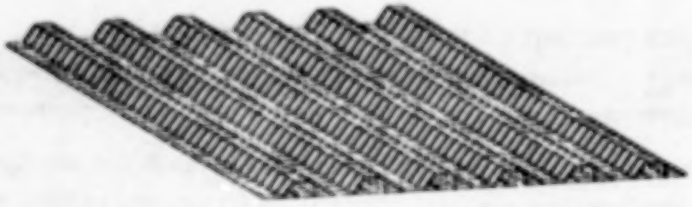
SPACE-ORIENTED APPLICATION STUDIES

Since the Challenger disaster, many of the CSM group have been involved with the recertification of the Solid Rocket Booster (SRB) for resumption of Space Shuttle flight. Various analyses have been performed to assess the overall structural response of the SRB. Structural analyses of the tang-clevis joints[3], detailed stress analyses of the SRB aft skirt, and nonlinear shell analyses of the SRB/ETA ring interface region[4] have been performed. These critical application problems have provided new goals for computational requirements in CSM and have contributed to solving an agency problem.

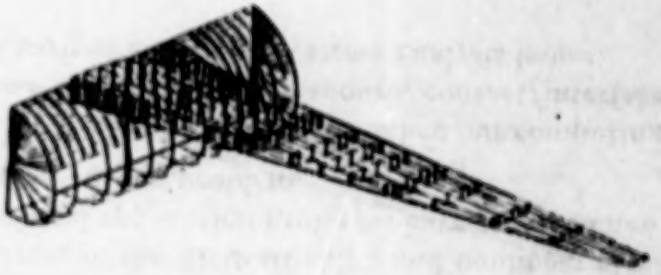
The solid rocket booster analyses have substantially challenged our structural analysis software and stretched our computing limits to redefine "a big nonlinear problem." The SRB problem encompasses nonlinear shell response, contact/interface problems, inelastic response, combined thermal and mechanical loadings, and various global/local stress analysis issues.

AERONAUTICS-ORIENTED APPLICATION STUDIES

GENERIC TRANSPORT



STIFFENED SHELLS



WING/FUSELAGE INTERSECTION

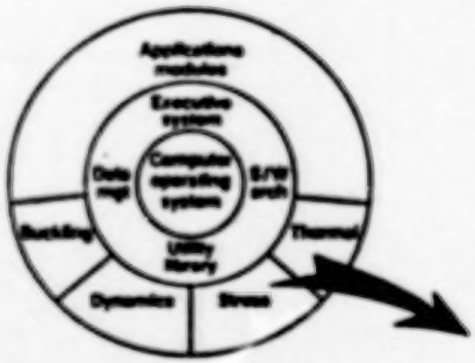
AERONAUTICS-ORIENTED APPLICATION STUDIES

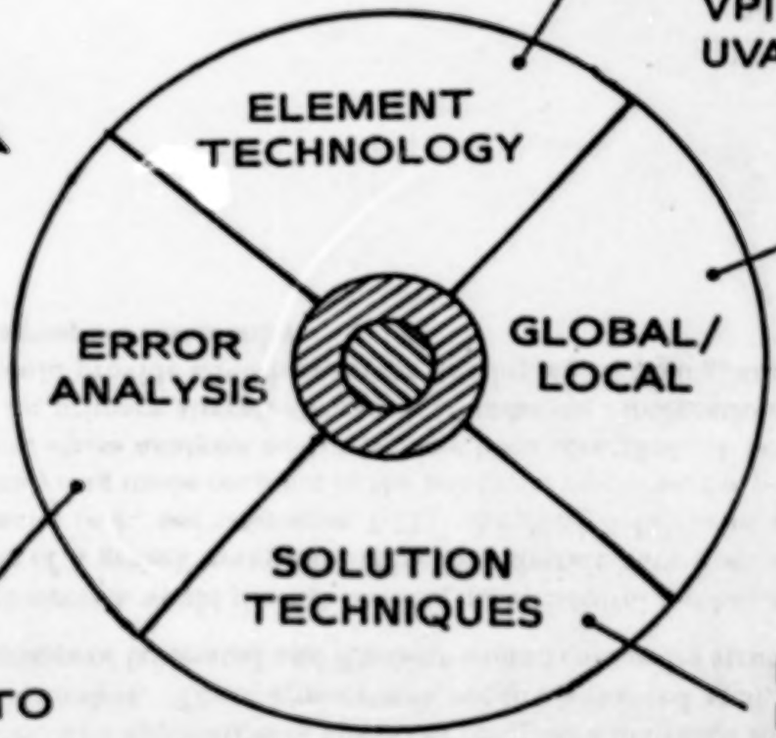
Generic transport aircraft structures offer another challenge to the computational structural analysts. Detailed geometry models of a transport severely tax today's computing environment.[5,6] Detailed response models of subscale components tax today's structural analysis tools. The application of advanced composite materials to aircraft structures has introduced new challenges to the designer and analyst. These applications require advanced analysis tools and an understanding of the response and failure characteristics of laminated and filament-wound composite structures.

A generic composite transport aircraft would present challenging structural mechanics problems as well as stretch our computing limits. Subcomponents of a generic composite transport aircraft have been studied experimentally, and the results correlated with analytical results (e.g., see references 7-11). Analysis deficiencies associated with the need for including transverse shear effects, for analyzing mode coupling in the nonlinear response, for performing a progressive failure analysis, and for performing global/local stress analyses routinely have been identified. A new NASA research initiative to develop advanced structural concepts for primary aircraft structures is underway. Integration of focus problems from this new initiative with the CSM program could provide a mechanism for developing, verifying, and transferring computational structural mechanics technology to the aerospace community.

ELEMENTS OF CSM

CURRENT METHODS RESEARCH THRUSTS





LaRC/CSM
LOCKHEED PALO ALTO
WASH. U.
VPI
UVA

LaRC/CSM
LOCKHEED PALO ALTO
VPI

LaRC/CSM,SMB
LOCKHEED PALO ALTO
U. COLORADO

LaRC/CSM,SMB
LOCKHEED PALO ALTO
U. TENN.

ELEMENTS OF CSM

Current Methods Research Thrusts

The CSM Testbed has been depicted as three concentric circles as shown in the upper left of the figure. The Methods and Application Studies Team develops application modules or processors and analysis procedures; that is, the outer circle. The elements of the Langley CSM program in the area of current methods research thrusts are shown in this figure.

The first area is element technology. New solid hybrid elements and flat shell hybrid elements are being developed by Dr. Mohammad Aminpour of the CSM Group. Under the CSM contract with Lockheed Palo Alto Research Laboratory (NASA Contract No. NAS1-18444), new curved shell elements are being developed by Drs. Gary Stanley and David Kang. In addition, a generic element processor has been designed by Lockheed to facilitate implementation, porting, and testing of elements in the CSM testbed. Under NASA Grant No. NAG-1-639, Dr. Barna Szabó of Washington University is developing hierarchic theories and element formulations. Under NASA Grant No. NAG-1-675, Dr. Hayden Griffin of Virginia Tech is implementing various displacement-based solid elements which have been developed as part of the NASA/Virginia Tech Composites Program. Under NASA Grant No. NGT-50116, Dr. Walter Pilkey of the University of Virginia has just begun the development of variationally-based element formulations.

The second area is global/local stress analysis. The CSM group is developing a 2-D global/local (coarse/refined) analysis capability. Under the CSM contract, Lockheed researchers are investigating the structural behavior of stiffened composite panels loaded in axial compression to assess the local stress state near the skin-stiffener interface region. In addition, Dr. Griffin of Virginia Tech is developing global/local stress analysis methodology for detailed, 3-D stress analysis of composite structures.

The third area is solution techniques. Lockheed researchers are involved in evaluating and implementing nonlinear solution strategies in the CSM Testbed. In addition, the Structural Mechanics Branch of the Structures and Dynamics Division at Langley is involved with advanced analysis techniques and demonstrating their capabilities using the STAGSC-1 computer code. A CSM objective is to incorporate that work in the Testbed. Under NASA Grant No. NAG-1-803, Dr. Alan George of the University of Tennessee is developing sparse matrix methods for serial and parallel computers and implementing these methods in the CSM Testbed.

The fourth area is error analysis. The CSM group is studying the strain energy gradient approach for guiding mesh refinement. In addition, Dr. Szabó of Washington University is studying p-extensions of the finite element method. Under NASA Grant No. NAG-1-802, Dr. John Dow of the University of Colorado at Boulder is developing an error estimation procedure based on the difference between the strain energy of the finite element solution and the strain energy of the "smoothed" solution.

FINITE ELEMENT TECHNOLOGY

- DEFICIENCIES -

- HEAVY RELIANCE ON FINITE ELEMENTS IN ANALYSIS AND DESIGN PROCESS

- EXISTING FINITE ELEMENTS SENSITIVE TO MESH DISTORTION

- STANDARD TEST PROBLEMS HAVE EXPOSED NUMEROUS ELEMENT DEFICIENCIES IN ALL FINITE ELEMENT CODES

- ELEMENT PERFORMANCE FOR NONLINEAR PROBLEMS HAS NOT BEEN ESTABLISHED

FINITE ELEMENT TECHNOLOGY

– Deficiencies –

The finite element method is over three decades old and continues to serve engineers as a powerful, general-purpose analysis tool for complex structures. The aerospace industry relies heavily on finite element analysis in the design and certification of aerospace structures. Finite element computer codes are readily available with a wide range of capabilities, cost, and user support. These analysis tools are often treated as "black boxes," and the developers of these tools assume that the user understands and works within the limitations of the software. Frequently, however, the limitations of the analysis are "extended" by novice users.

A common limitation in nearly all finite element computer codes is the sensitivity of the elements to mesh distortion (i.e., element warping, aspect ratio, element taper). Many of the finite element codes do perform checks to assess element geometry and inform the user of potential problems. Standard test problems for linear elastic stress analysis have been proposed.[12,13] These tests have exposed deficiencies in many elements. Standard test problems for nonlinear stress analysis are not well established, but are under development.

FINITE ELEMENT TECHNOLOGY

- CURRENT PROGRAM -

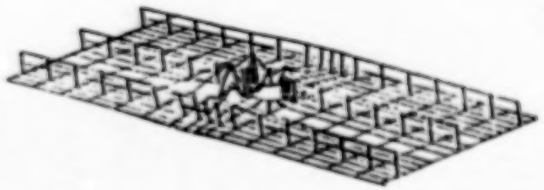
- GENERIC ELEMENT PROCESSOR PROVIDES COMMON "PROVING GROUNDS" FOR ELEMENT RESEARCH
- ADVANCED SHELL ELEMENT FORMULATIONS
- HIERARCHIC THEORIES AND ELEMENT FORMULATIONS
- ROBUST SOLID ELEMENTS FOR DETAILED ANALYSIS OF COMPOSITE STRUCTURES
- SPECIAL ELEMENTS FOR FRACTURE PROBLEMS

FINITE ELEMENT TECHNOLOGY
– Current Program –

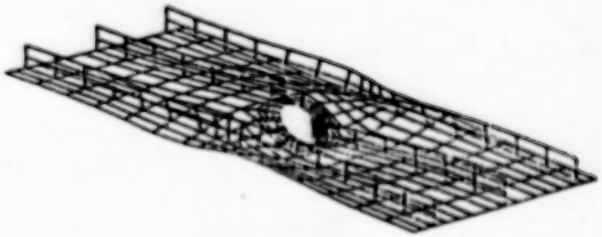
The current Langley CSM program in finite element technology has two main objectives. First, Lockheed researchers conceived and are now developing and implementing a new "generic element processor" which will enable routine assessment of new finite element formulations within the Testbed framework. Hence, the assessment will encompass not only standard test problems but also challenging focus problems, thereby establishing a "proving ground" or "obstacle course" for new elements. Second, advanced element formulations are being developed. Research topics include robust, nonlinear shell element formulations, p-version finite element technology, solid elements for detailed stress analysis of laminated composite structures, and special elements for fracture mechanics problems.

ASSESSMENT OF CSM TESTBED 2-D SHELL ELEMENTS

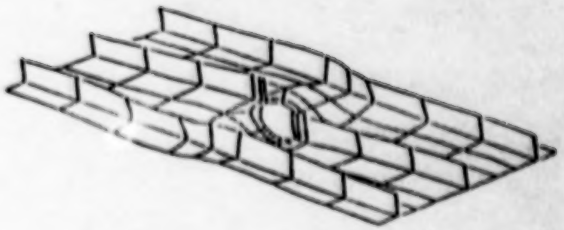
E43



EX47



EX97



- BUCKLING MODE SHAPE FOR FOCUS PROBLEM
- E43 ELEMENT SENSITIVE TO MESH DISTORTION
- EX47 AND EX97 ELEMENTS LESS SENSITIVE TO MESH DISTORTION AND INCORPORATE TRANSVERSE SHEAR

ASSESSMENT OF CSM TESTBED 2-D SHELL ELEMENTS

One of the initial focus problems of the CSM group is a composite blade-stiffened flat panel with a discontinuous stiffener. This problem has characteristics which often require a global/local stress analysis. These characteristics include a discontinuity, eccentric loading, large displacements, large stress gradients, high inplane loading, and a brittle material system. This problem represents a generic class of laminated composite structures with discontinuities in which the interlaminar stress state becomes important.

Finite element analysis of this focus problem tests the element's performance for a problem which requires a distorted mesh in order to model the region near the circular hole. Incorporating the effect of transverse shear deformation is needed to provide a 2-D assessment of the interlaminar stress state.

Linear stress analyses and bifurcation buckling analyses have been performed for this focus problem using various finite elements implemented in the CSM Testbed. The E43 element is the original hybrid stress C^1 flat shell element from SPAR Level 13.[14] The EX47 and EX97 elements are new C^0 shell elements based on the assumed natural strain formulation with four and nine nodes, respectively.[15]

The finite element grids used in these analyses have identical numbers of nodes, and therefore, each model has the same number of degrees-of-freedom. Each model predicts the stress distribution for the prebuckled stress state correctly. However, the results from the bifurcation buckling analyses differ. The models using the EX47 and EX97 elements agree with previously obtained results. The model using the E43 element does not predict the correct buckling load or mode shape. The buckling mode shape for each model is shown in the figure. These results indicate that the geometric stiffness matrix for the E43 element is sensitive to mesh distortion, while those for the EX47 and EX97 elements are not as sensitive. The EX47 and EX97 elements also incorporate transverse shear in their formulation.

GLOBAL/LOCAL STRESS ANALYSIS METHODOLOGY

- DEFICIENCIES -

- DESIGN PROCESS NEEDS ACCURATE, ROUTINE GLOBAL/LOCAL STRESS ANALYSIS CAPABILITY
- CURRENT APPROACH ASSUMES LOCAL REGION IS KNOWN A PRIORI
- NO AUTOMATED PROCEDURE
- ANALYST IDENTIFIES 2-D/3-D TRANSITION INTERFACE REGION BY TRIAL AND ERROR
- MODELING CRITERIA FOR DETAILED STRESS ANALYSIS INADEQUATE

GLOBAL/LOCAL STRESS ANALYSIS METHODOLOGY

– Deficiencies –

A myriad of definitions is associated with the term "global/local stress analysis." Herein global/local stress analysis methodology is defined as a procedure to determine local, detailed stress states for specific structural regions using information obtained from an independent global stress analysis. Furthermore, the global/local stress analysis methodology should not require a priori knowledge of the location of the local region(s) requiring special modeling. For example, the composite blade-stiffened panel with a discontinuous stiffener has an obvious local region which requires special modeling to predict accurately the stress state near the hole. However, for the curved composite panels described in reference 11, the local region requiring a detailed stress analysis is not obvious until after the global postbuckled solution is calculated.

The design and certification process for aerospace structures requires an accurate, routine global/local stress analysis capability. Several approaches are available in commercially available structural analysis codes like MSC/NASTRAN and ANSYS. These approaches include multilevel substructuring (e.g., see reference 16), spline interpolation along the local region boundaries, and transition grids (i.e., use of triangular elements) to refine locally near regions with large stress gradients. However, no automated procedure is available to the analyst for routine global/local stress analysis or to ensure that a continuous stress field results across a global-to-local transition boundary in cases where independent submodels are used. Transitioning from shell-to-solid elements is possible provided the analyst can identify the location of the 2-D/3-D transition interface region. Modeling criteria for detailed stress analysis are inadequate and require the analyst to perform numerous "pathfinder" studies to guide the modeling effort for each new application. General procedures and guidelines for 3-D stress analysis of composite structures are not readily available in the open literature.

GLOBAL/LOCAL STRESS ANALYSIS METHODOLOGY

- CURRENT PROGRAM -

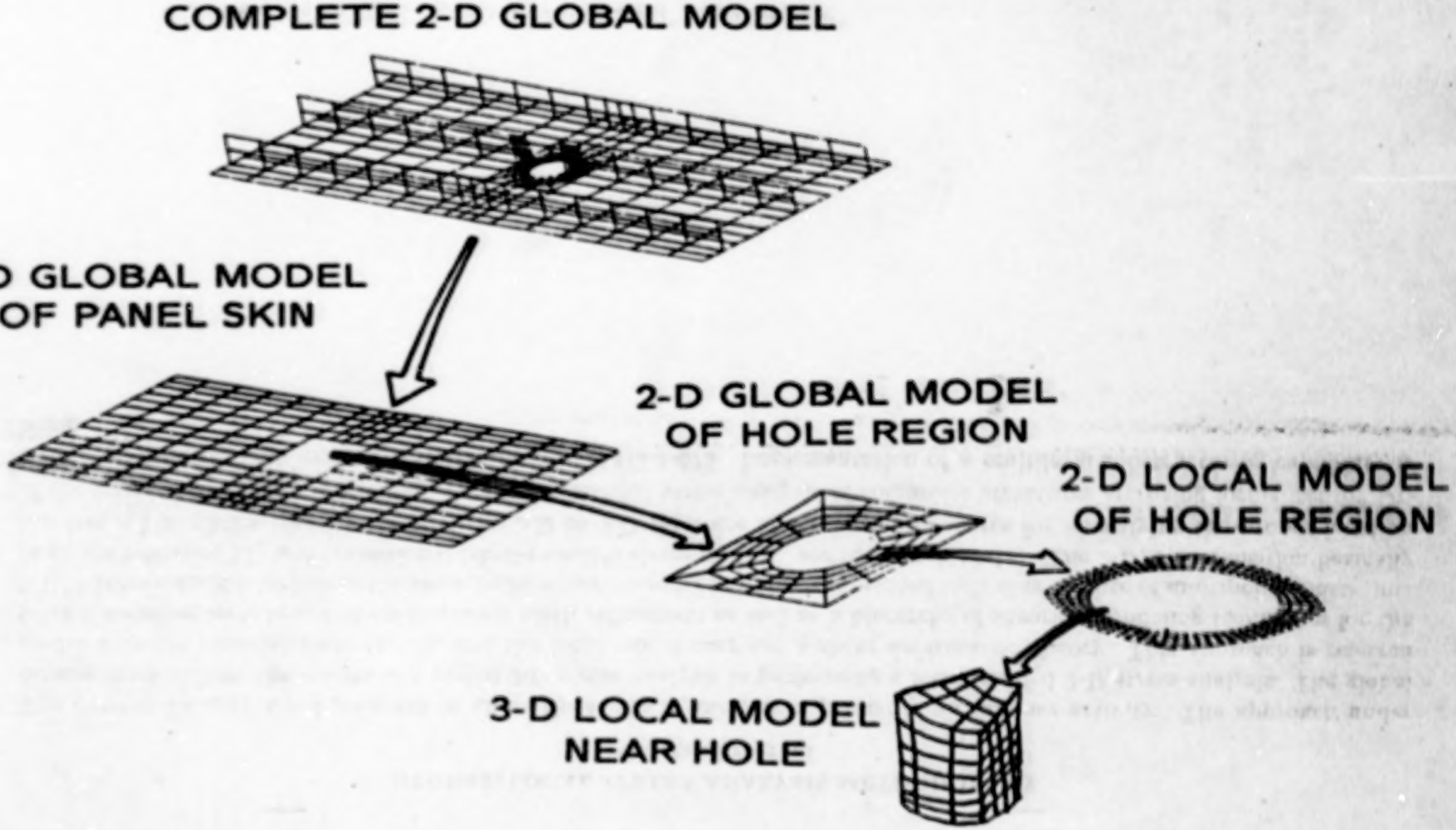
- HIERARCHIC THEORIES
  (CLASSICAL vs SHEAR FLEXIBLE vs 3-D)
- 2-D ZOOMING TECHNIQUES
- 2-D/3-D TRANSITION INTERFACE
- MODELING FOR DETAILED STRESS ANALYSIS OF COMPOSITE STRUCTURES
- TRANSITIONAL (SHELL-TO-SOLID) FINITE ELEMENTS
- MULTI-LEVEL SUBSTRUCTURING

GLOBAL/LOCAL STRESS ANALYSIS METHODOLOGY

– Current Program –

The current Langley CSM program in global/local stress analysis is primarily an inhouse activity. The approach under development utilizes the results of a global 2-D stress analysis in performing a local, refined 2-D stress analysis. The global model may use classical plate theory, and the local model may use a shear deformation theory. This approach is referred to as a zooming technique and encompasses mesh refinement as well as a hierarchy of theories. Modeling techniques for the 2-D/3-D transition interface in the same finite element model have been presented including the use of multipoint constraints (e.g., see reference 17) and transitional (shell-to-solid) elements (e.g., see references 18-19). The 2-D/3-D transition basically involves a kinematics transition across the 2-D to 3-D interface and robust procedures for identifying the proper location of the interface boundary. Modeling criteria for detailed stress analysis of composite structures are being formalized by Dr. Griffin of Virginia Tech under NASA Grant No. NAG-1-675. Implementation of a multilevel substructuring capability is being planned for the CSM Testbed.

"ZOOM-IN" APPROACH FOR GLOBAL/LOCAL STRESS ANALYSIS



"ZOOM-IN" APPROACH FOR GLOBAL/LOCAL STRESS ANALYSIS

The approach for global/local stress analysis is to predict the global nonlinear response using a complete, global 2-D model and then construct a refined, local 2-D model for a small distance away from the discontinuity to predict accurately the large stress gradient. Displacements and rotations from the global nonlinear solution obtained using the complete model are applied as boundary conditions to the refined model and the state of stress determined. This strategy is referred to as a multi-level or "zoom-in" approach. To establish the accuracy of the refined, local 2-D model near the discontinuity, a 3-D model is analyzed and the stress state determined.

This approach for global/local stress analysis does not require a priori knowledge as to the location of regions with a large stress gradient. Hence, a global nonlinear stress analysis may be performed, local regions of high stress identified, and local detailed stress analysis performed. The global nonlinear stress analysis using a refined mesh near the local regions of concern would not have to be performed.

SOLUTION TECHNIQUES

- DEFICIENCIES -

- VARIETY OF TECHNIQUES FOR LARGE DEFLECTION, LARGE ROTATION PROBLEMS
- LIMITED EVALUATION ON REALISTIC STRUCTURES
- NO WIDELY-ACCEPTED STANDARD TEST PROBLEMS
- OPTIMUM DESIGN WITH NONLINEAR RESPONSE BEYOND CURRENT COMPUTATIONAL CAPABILITIES
- PROBLEMS WITH MODE-INTERACTION BEYOND CURRENT NONLINEAR ANALYSIS CAPABILITIES

SOLUTION TECHNIQUES
– Deficiencies –

Solution techniques for nonlinear finite element analyses are emphasized in many research programs. A variety of techniques are available to solve nonlinear structural analysis problems involving large deflections and large rotations. Numerous papers and books (e.g., see references 20-22) are available that describe various techniques and demonstrate their application on structural problems. The evaluation of nonlinear solution techniques frequently involves problems with simple geometries; however, these problems usually embrace complex nonlinear response characteristics.

Standard test problems for nonlinear stress analysis are being developed, but no widely-accepted set of test problems has been adopted. Extending the evaluation of nonlinear solution techniques to realistic structures has received only limited attention to-date.

Optimum design of aerospace structures using nonlinear structural response is beyond current computational capabilities, primarily because of the computational cost of the nonlinear structural analyses. Optimized structural designs frequently result in a structure which has closely-spaced buckling loads. Consequently, predicting the nonlinear response of these structures may involve mode interaction, and thereby exceed the nonlinear analysis capabilities currently available in finite element computer codes.

SOLUTION TECHNIQUES

- CURRENT PROGRAM -

- ELEMENT-INDEPENDENT COROTATIONAL FORMULATION
  - LOW-ORDER
  - HIGHER-ORDER
- SOLUTION STRATEGIES FOR NONLINEAR PROBLEMS
  - NEWTON-RAPHSON METHOD
  - ARC-LENGTH CONTROL TECHNIQUES
  - NEWTON'S METHOD FOR ISOLATED BIFURCATION PROBLEMS
- SPARSE MATRIX METHODS

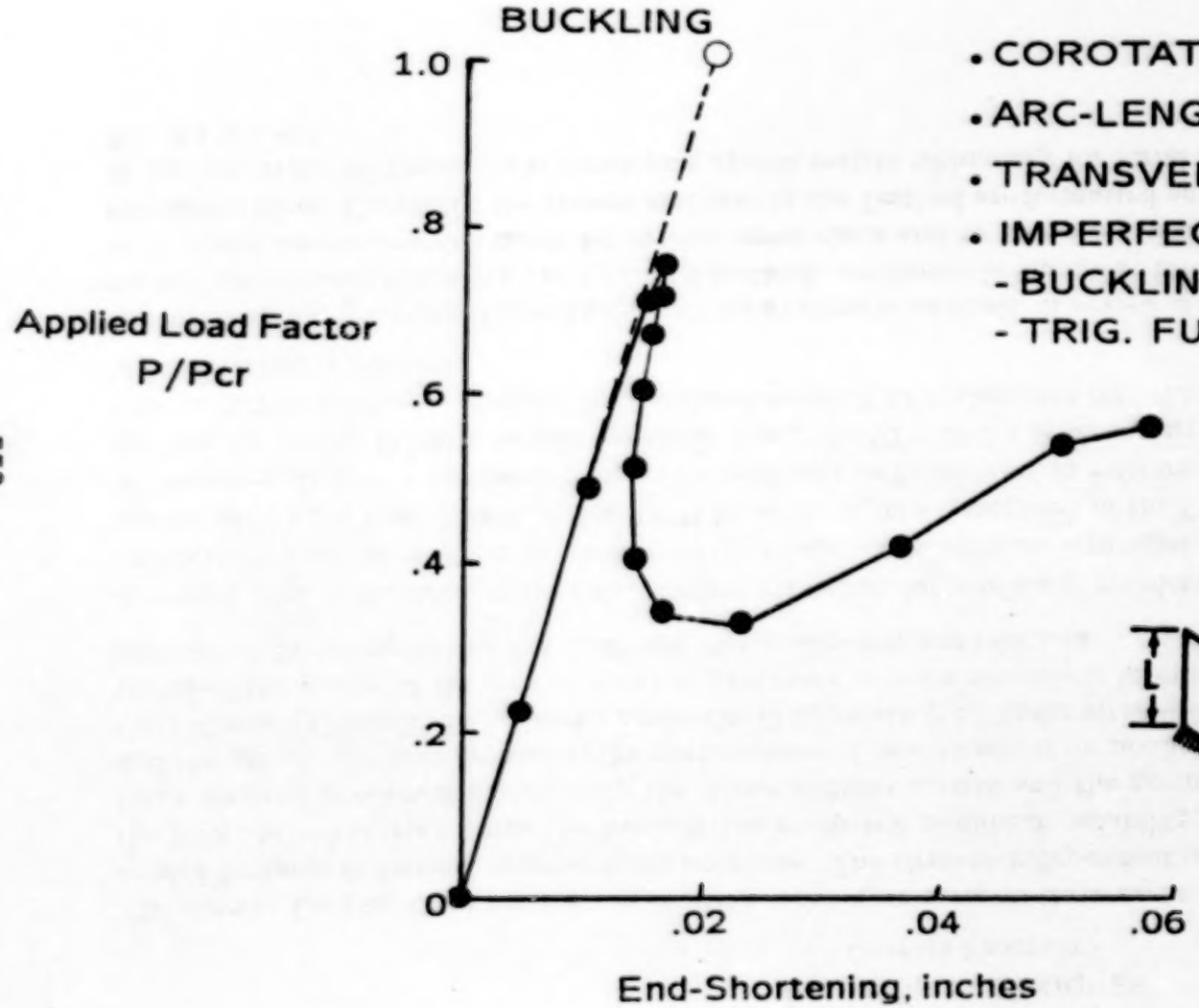
SOLUTION TECHNIQUES
– Current Program –

The current Langley CSM program in solution techniques involves three areas. One area addresses the formulation aspects needed for large deflection, large rotation problems. The element-independent corotational formulation [23,24] developed under the STAGSC-1 contract forms the basis of the geometric nonlinear capability of the finite elements in the CSM Testbed. Some element developers supply only the linear stiffness matrix and the geometric stiffness matrix. Using these matrices and the generic element processor, the performance of new elements on nonlinear problems may be readily assessed in the CSM Testbed through the low-order corotational approach (i.e., linear strain-displacement relations within the element). In addition, the design of the generic element processor includes provisions to implement and assess higher-order corotational approaches by incorporating the nonlinear strain-displacement relations.

A second area of research deals with solution strategies for nonlinear problems. The language capability of the Testbed architecture may be utilized to explore various aspects of solution strategies such as the Newton-Raphson method and various arc-length control techniques.[25,26,27] In addition, the framework of the CSM Testbed will enable exploratory studies of Newton's method for isolated bifurcation problems as formulated in reference 28. Fundamental research using Newton's method for nonlinear finite element analysis using the STAGSC-1 general purpose nonlinear shell finite element computer code is currently being performed by Lockheed under NASA Contract No. NAS1-18101.[24,29] Incorporating this technology in the Testbed is planned.

The third area involves matrix techniques for finite element analysis. A review of the requirements and needs of finite element researchers, numerical analysts, and parallel methods developers is underway through the Lockheed CSM contract. This task is to assess matrix analysis needs by various researchers and to propose a standard set of utilities and data structure for system matrices. Currently, the system matrices in the Testbed are formatted as node-oriented, sparse matrices. Dr. George of the University of Tennessee is developing sparse matrix technology for serial and parallel computers under NASA Grant No. NAG-1-803.

POSTBUCKLING RESPONSE OF ISOTROPIC CURVED PANEL DEMONSTRATED USING CSM TESTBED



- COROTATIONAL
- ARC-LENGTH CONTROL
- TRANSVERSE SHEAR
- IMPERFECTIONS
  - BUCKLING MODE
  - TRIG. FUNCTIONS

POSTBUCKLING RESPONSE OF ISOTROPIC CURVED PANEL DEMONSTRATED USING CSM TESTBED

To demonstrate the nonlinear analysis capability of the CSM Testbed, the postbuckling response of an isotropic curved panel loaded in axial compression is predicted. This analysis incorporates the corotational formulation, an arc-length control solution strategy implemented using the CLAMP language, and the effects of transverse shear deformation through the use of the new shell elements (EX97) which are based on an assumed natural-coordinate strain formulation. Initial geometric imperfections may be incorporated as either a linear combination of eigenvectors (buckling mode shapes) or a set of trigonometric functions (e.g., see reference 8).

ERROR ANALYSIS METHODOLOGY

- DEFICIENCIES -

- SEVERAL PROPOSED TECHNIQUES; NO GENERAL PURPOSE IMPLEMENTATIONS
- AD HOC APPROACH USED TO ASSESS OVERALL MODELING ACCURACY
- REQUIRES MULTIPLE SOLUTIONS FOR MULTIPLE MESHES TO ESTABLISH ACCURACY FOR LINEAR PROBLEMS
- CRITERIA FOR NONLINEAR PROBLEMS NOT AVAILABLE

ERROR ANALYSIS METHODOLOGY

– Deficiencies –

The aerospace industry relies heavily on finite element analysis for the design and certification of aerospace structures. However, specific requirements to certify the structural analysis and the structural analysis computer code are not imposed on the structural analyst. Conceivably, countless manhours could be used during a failure investigation that may have been avoided had an accurate and appropriate pre-test analysis been performed and certified.

Although considerable research has been performed and several techniques proposed for general purpose finite element computer codes, no general purpose implementations are available for error detection and control for finite element solutions. Presently, an ad hoc approach is used to assess overall modeling accuracy, element behavior, and solution quality. Multiple solutions for multiple meshes are required to establish solution accuracy for linear problems. Criteria for nonlinear problems have not been developed to-date. The analyst must assess the finite element solution and is responsible for insuring its accuracy.

ERROR ANALYSIS METHODOLOGY

- CURRENT PROGRAM -

- ERROR ESTIMATES FOR GUIDING ADAPTIVE MESH REFINEMENT
- GRADIENT OF STRAIN ENERGY DENSITY
- STRESS SMOOTHING; COMPARE ORIGINAL AND SMOOTHED VALUES
- ASSESS ERROR IN DISCRETE SOLUTION USING EQUILIBRIUM DIFFERENTIAL EQUATIONS
- p-VERSION FINITE ELEMENT METHOD
- "REPORT CARDS" FOR ELEMENT PERFORMANCE

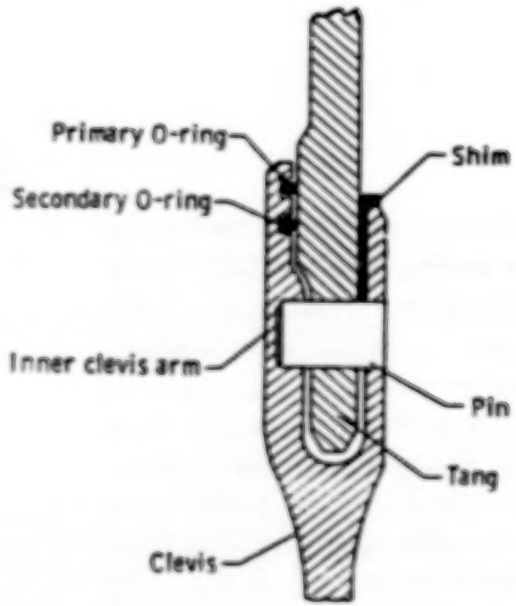
ERROR ANALYSIS METHODOLOGY

– Current Program –

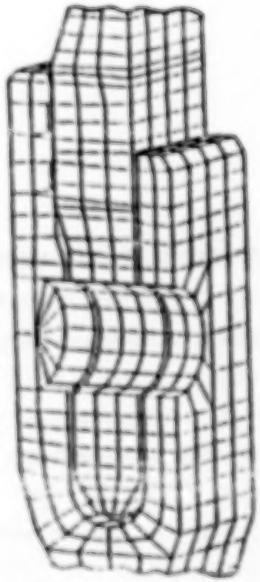
The current Langley CSM program in error analysis methodology is in its formative stage. Research in error detection and control for finite element analysis is a recent emphasis for the Methods and Application Studies Team. Given a discrete finite element solution, estimates of the error are needed to assess and improve solution quality and to provide heuristic guidelines for mesh refinement. Error estimates will be used to guide adaptive mesh refinement strategies and to provide measures of solution quality to the analyst. At present, a posteriori error estimates are being assessed in-house. Techniques associated with the strain energy, its gradient, and stress smoothing are being evaluated and implementation within the Testbed framework is planned. The Structural Mechanics Branch is evaluating the error in discrete finite element solutions by employing a "recontinuization" procedure and the nonlinear equilibrium differential equations.[30] Rigorous mathematical analyses have provided error estimates for the p-version finite element method, and convergence rates for the solution may also be calculated. In addition, numerous test problems are being solved; still many more have been proposed. These results may serve as "report cards" for element performance to the analyst by proving an assessment of finite element model convergence (e.g., see reference 31).

MESH DESIGN
FOR ACCURATE STRESS PREDICTION

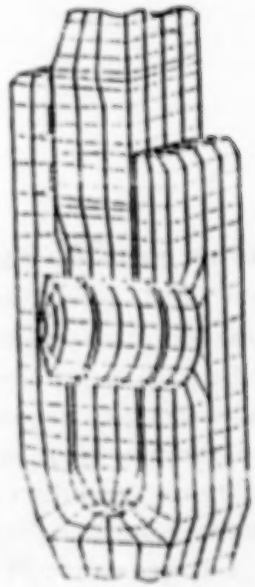
TANG-CLEVIS JOINT



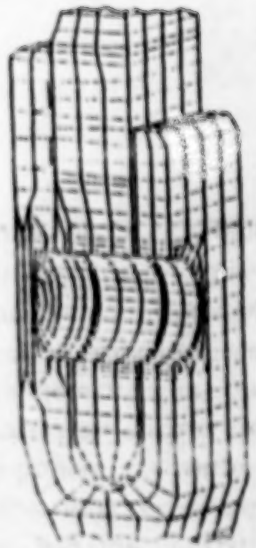
INITIAL



INTERMEDIATE



FINAL



MESH DESIGN FOR ACCURATE STRESS PREDICTION

Mesh design for complex structures requires the analyst to plan the analysis and to decide on the analysis objectives. An analysis to study only deflections has different requirements on spatial discretization than an analysis to determine the stress state. An example of such a problem is the tang-clevis joints on the Space Shuttle solid rocket motor shown on the left side of the slide. Relative motion between the inner arm of the clevis and the O-ring sealing surface on the tang has been identified as a potential contributor to the Challenger failure. Finite element structural analyses have been performed to predict both deflections and stresses in the joint under the primary, pressure loading condition. These analyses, reported in reference 3, demonstrate the difficulty of accurately predicting the structural behavior of the tang-clevis joint. To help understand the structural behavior of the tang-clevis joint, the sensitivity of the joint response to finite element mesh refinement in the area surrounding the pin is needed. The pin regions from three different finite element models are shown in the figure. There are two reasons for refining the initial model shown on left center side of the figure to the intermediate model shown in the right center of the figure. The first reason is to better account for pin flexibility. The second reason is to better model the bending of the inner clevis arm. The model refinement from the intermediate model shown in the right center to the final model shown on the right side of the figure is motivated by a need for a better prediction of gap motion and also for better stress predictions in the vicinity of the pin. A main feature of the final model is two additional rings of elements in the region where the pin contacts the inner clevis arm. These additional elements allow better modeling of the contact force distribution. The pin itself is also considerably more refined. Although the average gap motion did not vary much for these three discretizations, the predicted stress state near the pin regions did change significantly as the mesh was refined.

CSM TESTBED ENHANCEMENTS

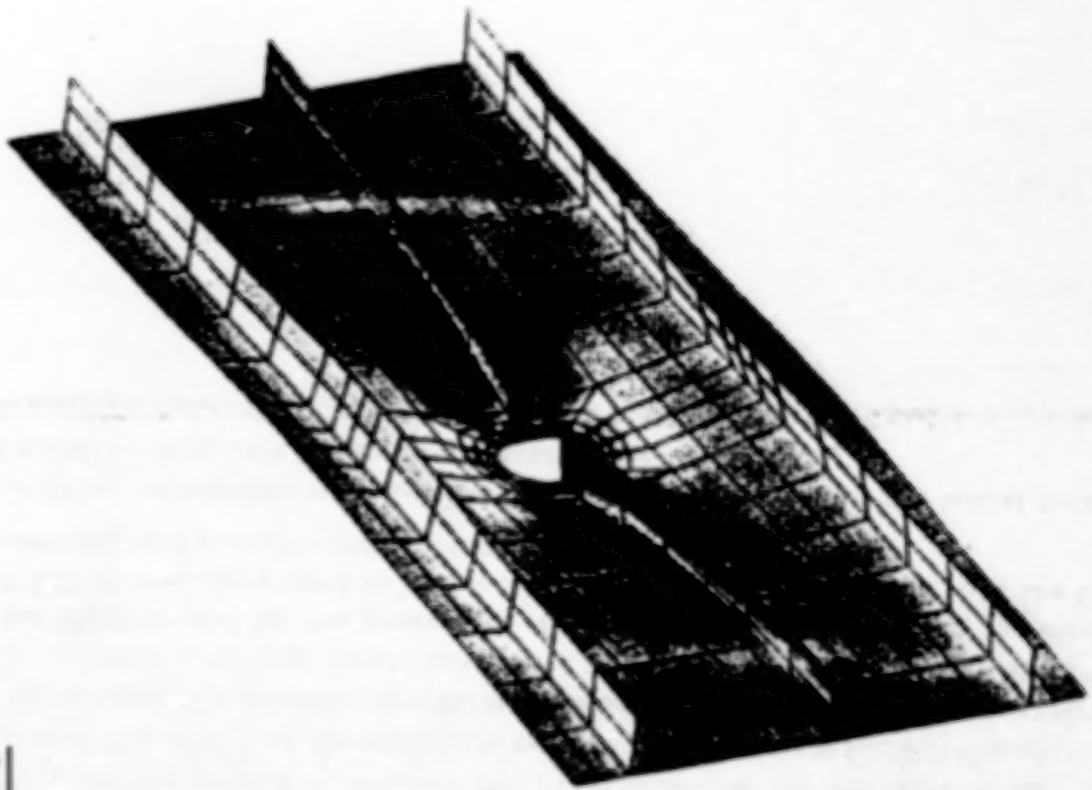
- RESEQUENCING ALGORITHMS FOR EQUATION SOLVER
- ARBITRARY STRESS REFERENCE FRAME FOR COMPOSITES
- LAMINATE ANALYSIS AND FIRST-PLY FAILURE UTILITIES
- PRELIMINARY 2-D GLOBAL/LOCAL STRESS ANALYSIS
- PRE- AND POST-PROCESSING INTERFACES FOR PATRAN

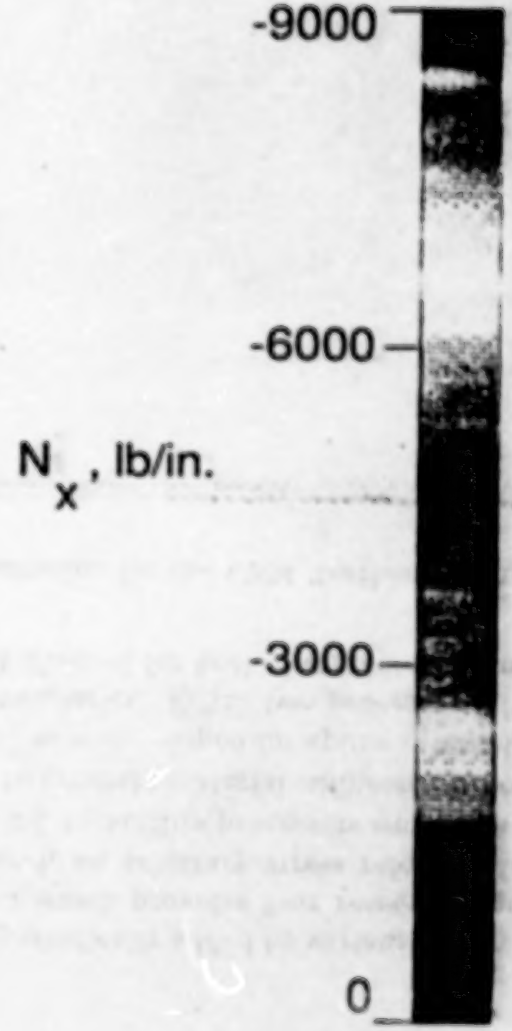
CSM TESTBED ENHANCEMENTS

Enhancing the CSM Testbed is an ongoing activity. New and improved features are periodically added by various researchers. Five major enhancements were added in FY87. First, a processor was developed which provides four resequencing (renumbering) algorithms to enhance the performance of the equation solver. Second, an arbitrary stress reference frame capability was added for the analysis of composite structures using 2-D elements. Third, two utility processors were developed for the analysis of composite structures. Processor LAU performs laminate analyses to generate material stiffnesses; processor FPF performs a first-ply failure analysis. Fourth, two processors (SPLN and INTS) were developed for spline interpolation of the solution and for the generation of applied displacements along the local boundaries. Fifth, two processors (PT2T and T2PT) were developed to interface PDA/PATRAN neutral files with the CSM Testbed for both pre-processing (model generation) and post-processing (results interpretation).

In addition, substantial emphasis has been placed on generating the initial documentation for the CSM Testbed. A draft of the CSM Testbed User's Manual has nearly been completed.

INPLANE STRESS RESULTANT N_x DISTRIBUTION ON DEFORMED STRUCTURE





ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

INPLANE STRESS RESULTANT N_x DISTRIBUTION ON DEFORMED STRUCTURE

The old adage that "a picture is worth a thousand words" is especially true in stress analysis of complex structures. The inplane stress resultant N_x distribution on the deformed geometry of the composite blade-stiffened panel with a discontinuous stiffener is shown in the slide. Color-coded stress contours can be extremely helpful to the structural analyst for determining the location of critical regions with high stresses.

ROLE OF CSM TESTBED

- SERVES AS FRAMEWORK FOR METHODS RESEARCH
  - GENERIC ELEMENT PROCESSOR FOR ELEMENTS
  - CLAMP LANGUAGE FOR SOLUTION ALGORITHMS
  - FPF PROCESSOR FOR FIRST-PLY FAILURE PREDICTION
- SERVES AS COMMON "PROVING GROUND" AND INTEGRATOR FOR METHODS RESEARCH
- WILL PROVIDE PROVEN, STATE-OF-THE-ART STRUCTURAL ANALYSIS METHODS TO AEROSPACE COMMUNITY

ROLE OF CSM TESTBED

The role of the CSM Testbed in methods research is threefold. First, the Testbed serves as a framework for methods research. The generic element processor enables easy, routine implementation of new finite element formulations and their evaluation. The CLAMP language provides a convenient mechanism for testing nonlinear solution strategies or direct time integration algorithms in a general purpose finite element code. The first-ply failure analysis provides composite structures analysts with information needed in design and test/analysis correlation studies. The CSM Testbed provides the opportunity to conduct advanced structural analysis research that exploits new and future computer hardware developments.

Second, the Testbed serves as a common "proving ground" and integrates various CSM researchers. Assessment of finite element technology, equation solvers and eigensolvers, nonlinear solution strategies, and other areas of methods research may be performed in a single, common software system, namely the CSM Testbed.

Third, the Testbed will provide proven, state-of-the-art structural analysis methods to the aerospace community. New methods will mature in the Testbed through application to selected challenging focus problems.

SUMMARY

- EMPHASIZING APPLIED STRUCTURAL MECHANICS RESEARCH TO ACCELERATE METHODS TECHNOLOGY TRANSFER TO INDUSTRY

- DEVELOP PROBLEM-ADAPTIVE SOLUTION STRATEGIES WITH ERROR CONTROL FOR **ROUTINE** GLOBAL/LOCAL STRESS ANALYSIS

- INCLUDE REALISTIC COMPLEX STRUCTURES AS FOCUS PROBLEMS TO EVALUATE NEW METHODS AND TO DEFINE COMPUTATIONAL STRUCTURAL ANALYSIS REQUIREMENTS

SUMMARY

The Langley CSM methods research efforts emphasize applied structural mechanics research to accelerate the transfer of methods technology to industry. Structural analysis and computational methods that have reached a level of maturity to demonstrate high potential for solving realistic, practical structural problems will continue to be a focus.

A goal is to develop problem-adaptive solution strategies with error control for routine global/local stress analysis. To meet this goal, four research areas have been identified. These research areas include finite element technology, solution techniques, global/local stress analysis methodology, and error analysis methodology.

As methods mature and computational structural mechanics technology develops into software, the focus problems will change to continue challenging our structural analysis capabilities and stretching our computing limits. Realistic complex structures like an SRB or a generic composite transport aircraft are needed as focus problems to evaluate new methods and to define new computational structural analysis requirements.

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES

1. National Research Council: *Computational Mechanics: A Perspective on Problems and Quality of Engineering*, NTIS Document Number PB85-128106, October 1984.

2. Knight, Norman F., Jr.; and Stroud, W. Jefferson: *Computational Structural Mechanics: A New Activity at the NASA Langley Research Center*, NASA TM-87612, September 1985.

3. Greene, William H.; Knight, Norman F., Jr.; and Stockwell, Alan E.: *Structural Behavior of the Space Shuttle SRM Tang-Clevis Joint*, NASA TM-89018, September 1986.

4. Knight, Norman F., Jr.: *Nonlinear Shell Analyses of the SRB/ETA Ring Interface*, NASA TM-89164, July 1987.

5. Rommel, R. A.; and Dodd, A. J.: Practical Considerations in Aeroelastic Design. *Recent Experiences in Multidisciplinary Analysis and Optimization*, NASA CP-2327, Part 1, April 1984, pp. 35-46.

6. McClellan, Gisela W.: The Structural Finite Element Model of the C-5A. *Thirteenth NASTRAN Users' Colloquium*, NASA CP-2373, 1985, pp. 364-376.

7. Starnes, James H., Jr.; Dickson, John N.; and Rouse, Marshall: Postbuckling Behavior of Graphite-Epoxy Panels. *ACEE Composites Structures Technology: Review of Selected NASA Research on Composite Materials and Structures*, NASA CP-2321, 1984, pp. 137-160.

8. Knight, Norman F., Jr.; and Starnes, James H., Jr.: Postbuckling Behavior of Selected Curved Stiffened Graphite-Epoxy Panels Loaded in Axial Compression. AIAA Paper No. 85-0768-CP, April 1985.

9. Starnes, J. H., Jr.; Knight, N. F., Jr.; and Rouse, M.: Postbuckling Behavior of Selected Flat Stiffened Graphite-Epoxy Panels Loaded in Compression. *AIAA Journal*, vol. 23, no. 8, August 1985, pp. 1236-1246.

10. Knight, N. F., Jr.; and Starnes, J. H., Jr.: Postbuckling Behavior of Axially Compressed Graphite-Epoxy Cylindrical Panels with Circular Holes. *Journal of Pressure Vessel Technology*, vol. 107, November 1985, pp. 394-402.

11. Knight, Norman F., Jr.; Starnes, James H., Jr.; Waters, W. Allen: Postbuckling Behavior of Selected Graphite-Epoxy Cylindrical Panels Loaded in Axial Compression. AIAA Paper No. 86-0881-CP, May 1986.

12. MacNeal, Richard H.; and Harder, Robert L.: A Proposed Standard Set of Problems to Test Finite Element Accuracy. *Journal of Finite Elements in Analysis and Design*, vol. 1, 1985, pp. 3-20.

13. Belytschko, T.; and Liu, W. K.: Test Problems and Anomalies in Shell Finite Elements. *Reliability of Methods for Engineering Analysis*, Edited by K. J. Bathe and D. R. J. Owen, Pineridge Press International, Swansea, U.K., 1986, pp. 393-406.

14. Pian, T. H. H.: Derivation of Element Stiffness Matrices by Assumed Stress Distributions. *AIAA Journal*, vol. 2, 1964, pp. 1333-1336.

15. Stanley, G. M.; Park, K. C.; and Hughes, T. J. R.: Continuum-Based Resultant Shell Elements. *Finite Element Methods for Plate and Shell Structures, Volume 1: Element Technology*, Edited by T. J. R. Hughes and E. Hinton, Pineridge Press International, Swansea, U.K., 1986, pp. 1-45.

16. Wilkins, D. J.: A Preliminary Damage Tolerance Methodology for Composite Structures. *Failure Analysis and Mechanisms of Failure of Fibrous Composite Structures*, NASA CP-2278, 1983, pp. 67-93.

17. Feld, D. J.; and Soudry, J. G.: Modeling the Interface Between Shell and Solid Elements. *MSC/NASTRAN Users' Conference Proceedings*, March 24-25, 1983.

18. Bathe, K. J.: *Finite Element Procedures in Engineering Analysis*. Prentice Hall, Inc., Engelwood Cliffs, NJ, 1982, pp. 259-260.

19. Surana, K. S.: Transition Finite Elements for Three-Dimensional Stress Analysis. *International Journal for Numerical Methods in Engineering*, vol. 15, July 1980, PP. 991-1020.

20. Stricklin, J. A.; and Haisler, W. E.: Formulations and Solution Procedures for Nonlinear Structural Analysis. *Computers and Structures*, vol. 7, 1977, pp. 125-136.

21. Bergan, P. G.: Solution Algorithms for Nonlinear Structural Problems. *Computers and Structures*, vol. 12, 1980, pp. 497-509.

22. Hinton, E.; Owen, D. R. J.; and Taylor, C.: *Recent Advances in Nonlinear Computational Mechanics*. Pineridge Press Limited, Swansea, U. K., 1982.

23. Rankin, C. C.; and Brogan, F. A.: An Element-Independent Corotational Procedure for the Treatment of Large Rotations. *Journal of Pressure Vessel Technology*, vol. 108, 1986, pp. 165-174.

24. Rankin, C. C.; Stehlin, P.; and Brogan, F. A.: *Enhancements to the STAGS Computer Code*, NASA CR-4000, November 1986.

25. Riks, E.: *Bifurcation and Stability, A Numerical Approach*. National Aerospace Laboratory NLR, the Netherlands, Report No. NLR-MP-84078U, 1984.

26. Crisfield, M. A.: An Arc-Length Method Including Line Searches and Acceleration. *International Journal for Numerical Methods in Engineering*, vol. 19, 1982, pp. 1269-1289.

27. Crisfield, M. A.: A Fast Incremental/Iterative Solution Procedure that Handles Snap-Through. *Computers and Structures*, vol. 13, 1983, pp. 55-62.

28. Thurston, G. A.: Newton's Method: A Link Between Continuous and Discrete Solutions of Nonlinear Problems. *Research in Nonlinear Structural and Solid Mechanics*, NASA CP-2147, 1980, pp. 27-46.

29. Thurston, G. A.; Brogan, F. A.; and Stehlin, P.: Postbuckling Analysis Using a General-Purpose Code. *AIAA Journal*, vol. 24, no. 6, June 1986, pp. 1013-1020.

30. Thurston, Gaylen A.; Reissner, John E.; Stein, Peter A.; and Knight, Norman F., Jr.: Error Analysis and Correction of Discrete Solutions from Finite Element Codes. AIAA Paper No. 84-0940, May 1984.

31. Verma, A.; and Melosh, R. J.: Numerical Tests for Assessing Finite Element Model Convergence. *International Journal for Numerical Methods in Engineering*, vol. 24, 1987, pp.843-857.

N89
29795
UNCLAS

GENERIC ELEMENT PROCESSOR

(APPLICATION TO NONLINEAR ANALYSIS)

by

Gary Stanley
Lockheed Palo Alto Research Laboratory
Computational Mechanics Section

ORIGINAL PAGE IS
OF POOR QUALITY

ABSTRACT

The CSM software Testbed, currently under joint development by NASA Langley (LaRC), Lockheed Palo Alto Research Laboratory (LPARL) and selected University Grantees, will provide researchers with a common workbench to develop computational methods for advanced structural analysis problems (such as the NASP and Space Station); and to apply these methods to large-scale components without major software implementation changes. The most novel aspect of this Testbed is that it is designed to sustain both parallel development (by Government, University and Industry) and parallel-processing computers.

The focus ~~of the present talk~~ is on one aspect of the ~~CSM~~ Testbed: finite element technology. The approach involves a 'Generic Element Processor': a command-driven, database-oriented software shell that facilitates introduction of new elements into the testbed. This 'shell' features an element-independent corotational capability that upgrades linear elements to geometrically nonlinear analysis, and corrects the rigid-body errors that plague many contemporary plate and shell elements. Specific elements that have been implemented in the Testbed via this mechanism include the Assumed Natural-coordinate Strain (ANS) shell elements, developed with Professor K.C. Park (University of Colorado, Boulder), a new class of curved hybrid shell elements, developed by Dr. David Kang of LPARL (formerly a student of Professor T. Pian), other shell and solid hybrid elements developed by NASA personnel, and recently a repackaged version of the workhorse shell element used in the traditional STAGS nonlinear shell analysis code.

The presentation covers (i) user and developer interfaces to the generic element processor, (ii) an explanation of the built-in corotational option, (iii) a description of some of the shell-elements currently implemented, and (iv) application to sample nonlinear shell postbuckling problems. There will also be a brief overview of the CSM Testbed: its current status, future directions, and potential benefits for all involved with the CSM effort.

N89-29795

OUTLINE

- PROBLEM: CSM ("Chaotic" Structural Mechanics)
- APPROACH: The Role of the Testbed
- PROGRESS: Testbed Development Overview
- GENERIC ELEMENT PROCESSOR
  - — Why? (Motivation)
  - — What? (User's View; Developer's View)
  - — How? (Nonlinear Applications)
- CONCLUSIONS
  - — Summary
  - — Pitfalls
  - — Plans

OUTLINE

The presentation begins with a curious look at the state of this so-called "mature" technology called computational structural mechanics (CSM). After motivating the need for an end to the present chaos, the CSM Testbed is presented as a viable solution that will enable us to approach next generation structural problems in concert. Next, an overview of the Testbed's development is presented, as background for the recent development of a Generic Element Processor. The Generic Element Processor is then discussed from all angles: why it was developed, how it should make life easier for the user, the algorithm developer, the element developer; and, finally, a demonstration of its application to nonlinear shell postbuckling analysis. The talk concludes with a summary of progress, a list of pitfalls, and specific plans for the next stage of development.

"Chaotic" Structural Mechanics (1)

FINITE ELEMENTS PROVIDE CONVENIENT BUILDING BLOCKS

but:

- **Why are there hundreds of different types?**
- **Why does each have a pathology?**
- **How do we select an element for an application?**
- **How do we adaptively refine a model — for nonlinear analysis?**

"Chaotic" Structural Mechanics (1)

Lockheed Palo Alto

CSM Testbed Development

Finite elements have been successful for very good reasons. One is that they provide convenient building blocks — both conceptually and in software — for general purpose structural analysis. However, why do we have hundreds of different element types when only a few intrinsic types are needed (e.g., beam, shell, solid, constraint)? Also with the plethora of different variations on a theme, it is disturbing that all elements seem to have at least one pathology (e.g., locking, spurious modes, distortion sensitivity). Thus, the *number* of different elements is not necessarily a sign of maturity. Given this situation, it becomes quite difficult to select an element for a given application. Finally, while techniques for discrete error estimation and adaptive mesh refinement are becoming a reality, there is a long way to go before such techniques can be applied to nonlinear analysis.

ORIGINAL PAGE IS
OF POOR QUALITY

"Chaotic" Structural Mechanics (2)

NONLINEAR SOLN ALGORITHMS CAN TRAVERSE LIMIT POINTS

but:

- **Why do they get stuck at bifurcations?**
- **Why are some sensitive to physical units?**
- **How do we pick an error tolerance?**
- **How do we parallelize them?**
- **How many iterations will it take to implement?**

"Chaotic" Structural Mechanics (2)

Another CSM area where significant progress has been made is nonlinear solution algorithms. Static continuation methods, such as arc length control, now make it possible to traverse limit points with ease. However, most practical shell problems involve bifurcations, which can lead to a variety of potential equilibrium solutions, and simple continuation methods do not provide a means for finding such paths. While research in this area is progressing rapidly, results are embryonic and there is some indication that *dynamic* analysis (or dynamic relaxation) may be the only reliable way to solve complex shell postbuckling problems. More work is also needed to make the static continuation methods now in use more robust. For example, the success of some algorithms may actually depend on the choice of physical units — due to scaling problems between displacements and forces and/or between translational and rotational degrees of freedom. Moreover, the selection of an error tolerance for nonlinear convergence remains a "black art"; in fact even the definition of an error norm can be plagued with the same kind of scaling problems as arise for continuation methods. Finally, we are just beginning to look at algorithms for solving nonlinear problems faster — on parallel computers. This may may add yet another dimension to the reliability hurdles mentioned above.

ORIGINAL PAGE IS
OF POOR QUALITY

"Chaotic" Structural Mechanics (3)

"GENERAL PURPOSE" F.E. PROGRAMS ABOUND!

but:

- **Why is there always something missing?**
- **Why is it so hard to add it?**
- **Why must we pay to use black boxes?**
- **Why can't we use the same code for research AND production?**
- **How will they survive the parallel processing era?**

"Chaotic" Structural Mechanics (3)

The last indicator of the "state of our art" to be looked at here is the availability of CSM software. While general-purpose structural finite element programs abound (and to a lesser extent those based on boundary elements, finite differences, etc.) there is little indication that they are doing anything to advance technology. They represent an assortment of black boxes, with something missing from each, and usually require great difficulty to add anything new. What's more, we are forced to pay for these technological straight jackets — or to develop our own private software at much greater cost. Haven't we learned enough during the past two decades to develop modular, extendible, white-box software that can be used by both researchers *and* production analysts?

ORIGINAL PAGE IS
OF POOR QUALITY

The Role of the CSM Testbed

Lockheed Palo Alto *CSM Testbed Development*

An

EXTENDIBLE

PUBLIC DOMAIN

TOOL KIT

for

PARALLEL DEVELOPMENT

and

PARALLEL PROCESSING

of

ADVANCED CSM SOLUTIONS

The Role of the CSM Testbed

The CSM Testbed is an approach for obtaining advanced solution to advanced structural mechanics problems by overcoming the obstacles described in the preceeding charts. It is an extendible, public domain software tool kit (or "operating system" if you like) designed for both parallel development and parallel processing, and for applications ranging from academic research on new analysis methods to production analysis of complex aerospace vehicles. The next few charts describe its evolution over the past few years from a set of independent software utilities to what is approaching an integrated programming environment.

ORIGINAL PAGE IS
OF POOR QUALITY

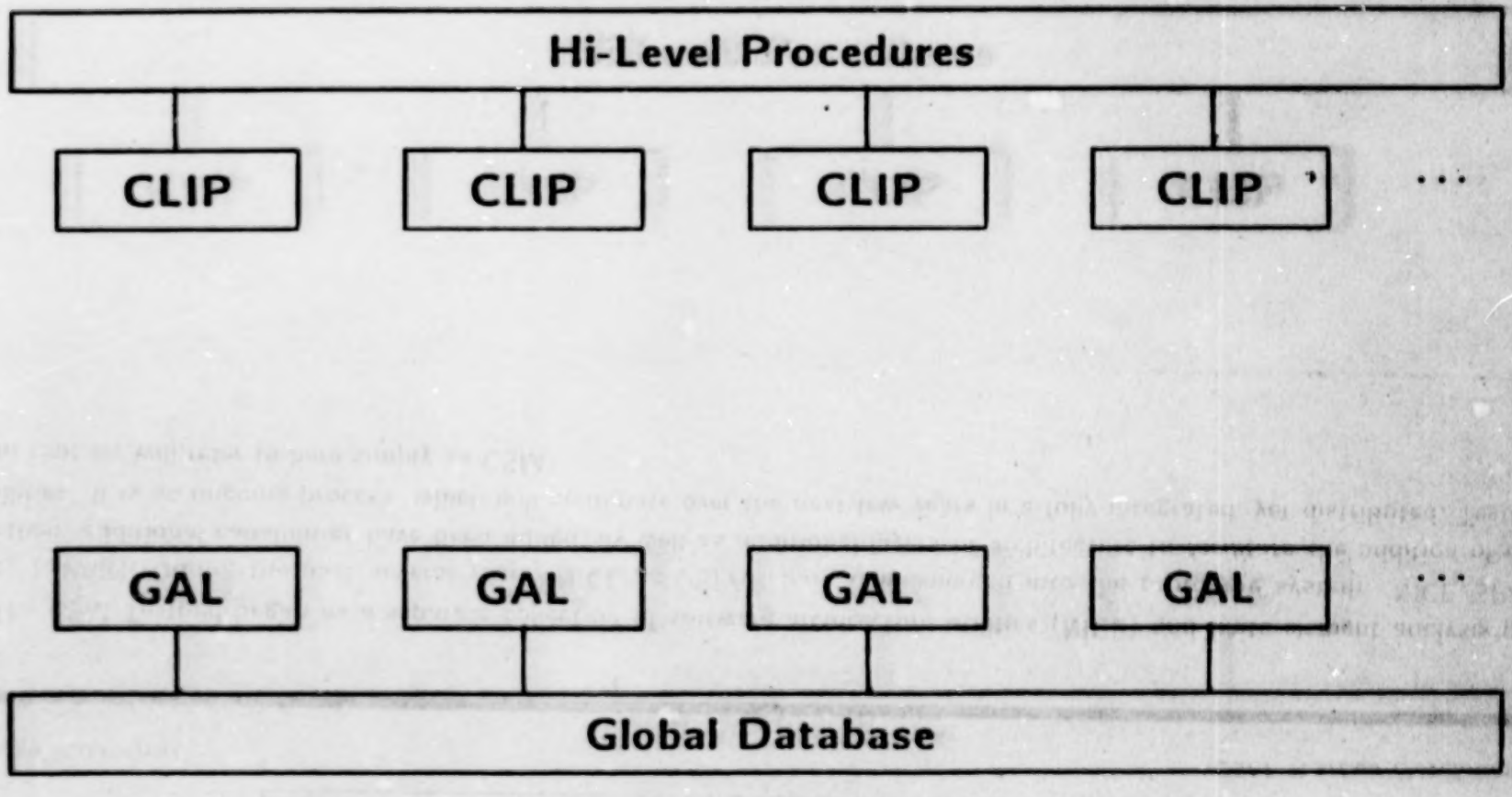
TESTBED DEVELOPMENT OVERVIEW

- **PHASE 0 — NICE and SPAR**
- **PHASE 1 — NICE/SPAR**
- **PHASE 2 — CSM**

TESTBED DEVELOPMENT OVERVIEW

The CSM Testbed began as a separate collection of software architecture utilities (NICE) and finite element analysis processors (SPAR). During the past several years, NICE and SPAR have been merged into the prototype system: NICE/SPAR. Since then, additional capabilities have been added, as well as additional layers of architecture to *facilitate* the addition of new capabilities. It is an ongoing process, which will culminate over the next few years in a fully integrated, yet distributed, Testbed system that we will refer to here simply as CSM.

PHASE 0: NICE
(Architecture Tools)

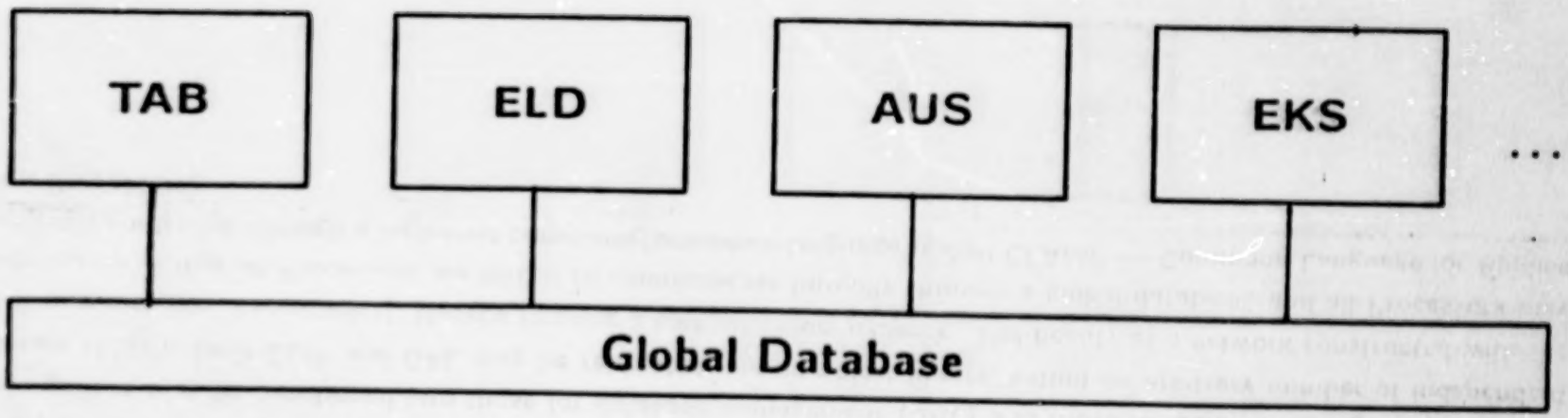


PHASE 0: NICE
(Architecture Tools)

Lockheed Palo Alto *CSM Testbed Development*

NICE (Network of Integrated Computational Elements) can be viewed as a set of software architectural utilities for CSM [1]. These utilities may be partitioned into those for database management (GAL) and those for command language/procedure interpretation (CLIP). Both CLIP and GAL may be replicated, via an object library, within an arbitrary number of independent software Processors (i.e., executables), thereby forming a computational network. The beauty of a network constructed with the NICE architecture is that all Processors are forced to communicate formally through a global database, and all Processors may be conveniently controlled through a high-level command/procedure language (called CLAMP — Command Language for Applied Mechanics Processors).

PHASE 0: SPAR
(Linear Analysis)

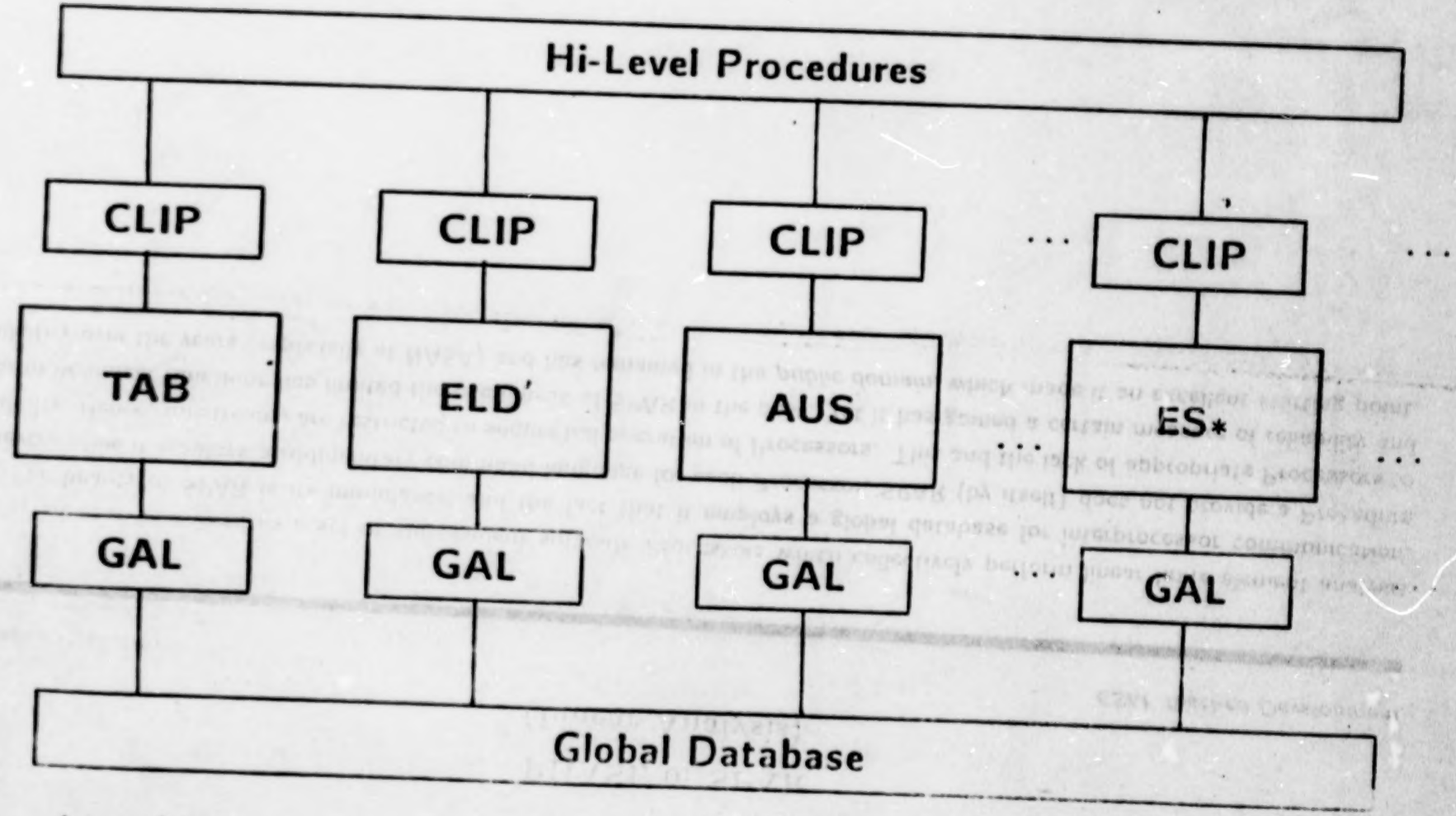


PHASE 0: SPAR
(Linear Analysis)

Lockheed Palo Alto *CSM Testbed Development*

SPAR may be viewed as a set of independent software Processors which collectively perform linear finite element analysis [2]. The beauty of SPAR is its modularity, and the fact that it employs a global database for interprocessor communication. However, while it employs a rudimentary command language for each Processor, SPAR (by itself) does not provide a Procedure capability. Hence, runstreams are restricted to sequential operation of Processors. This and the lack of appropriate Processors to perform nonlinear functions has limited the usefulness of SPAR in the past. Yet it has gained a certain measure of reliability and familiarity over the years (especially at NASA) and has remained in the *public domain*, which made it an excellent starting point.

PHASE 1: NICE/SPAR
(Prototype)



PHASE 1: NICE/SPAR
(Prototype)

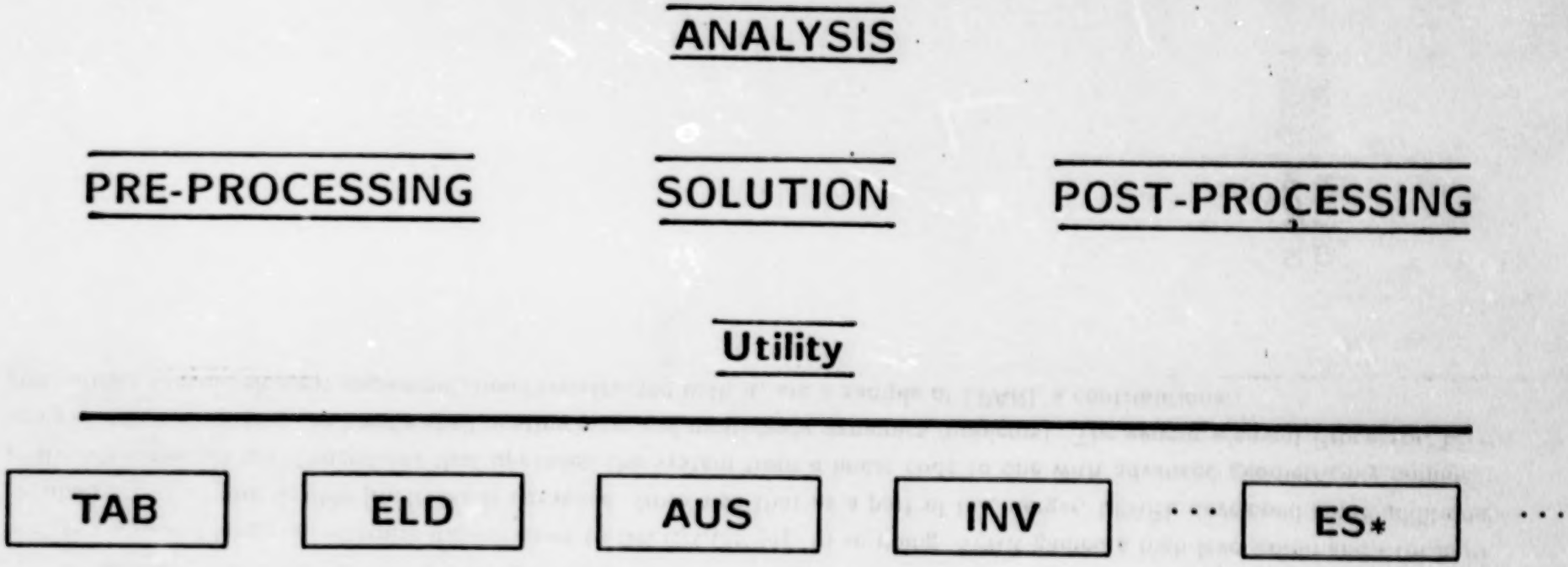
Lockheed Palo Alto *CSM Testbed Development*

An obvious match. NICE and SPAR were mated by interfacing SPAR's database management utilities to NICE/GAL [3], and its command language interpretation utilities to NICE/CLIP [4]. In so doing, SPAR gained a high-level command Procedure capabilitiy and a more flexible (multi-level) database. Note also that as a part of the merger, LPARL developed some additional analysis Processors and Procedures that upgraded the system from a linear code to one with advanced geometrically nonlinear analysis capabilities (e.g., to handle shell postbuckling and multi-body dynamics problems). The generic element Processor, ES*, and various specific element implementations constructed with it, are a sample of LPARL's contributions.

ORIGINAL PAGE IS
OF POOR QUALITY

NICE/SPAR Hi-Level Procedures

Lockheed Palo Alto *CSM Testbed Development*



NICE/SPAR Hi-Level Procedures

The next few charts illustrate how the NICE Procedure level was exploited in NICE/SPAR to conceal the details of individual Processor execution, permiting the user to access the system with an appropriate level of abstraction. Thus, very high-level Analysis Procedures were written (in the CLAMP language) which in-turn invoke generic model (Pre-processing) Procedures, linear and nonlinear Solution Procedures, and an assortment of Post-processing Procedures (as shown in this chart). Note that direct execution of individual Processors (shown below the horizontal line) is still permitted; high-level Procedures simply provide a tool that can be quickly and easily tailored to the application — be it research- or production-driven.

ORIGINAL PAGE IS
OF POOR QUALITY

NICE/SPAR

Pre-Processing Procedures


```
*procedure GEN_CYLINDER (
                elt_proc = ES1 ;  elt_type = ANS9 ;  elt_pars = 0.0
                radius = 1.0 ; thickness = .1 ; E = 1.e7 ; PR = .3
                bc_procedure = PINCHED_CYL_BC
                nel_x = 3 ; nel_y = 3 )
    [xqt TAB
    *call ES ( function = 'DEFINE ELEMENTS' )
    [xqt ELD
    [xqt AUS
    [xqt E
    [xqt TOPO
*end
```

NICE/SPAR

Pre-Processing Procedures

Lockheed Palo Alto *CSM Testbed Development*

Shown in this chart is an example (in outline form) of a CLAMP Procedure to define a generic cylindrical shell finite element model, using NICE/SPAR Processors. The example merely illustrates the nature of Procedure arguments (they are name-lists, allow default settings and are order-independent), shows the incorporation of SPAR Processor executions (e.g., |xqt TAB), and also shows the invocation of one Procedure from another: in this case, GEN_CYLINDER calls ES, which, as we shall see, is a multi-purpose window to the Generic Element Processor.

ORIGINAL PAGE IS
OF POOR QUALITY

NICE/SPAR

Linear Solution Procedures


```
*procedure L_STATIC_1 ( )

  *call ES ( function = 'FORM STIFFNESS/MATL' )
  [xqt K
  [xqt INV
  [xqt SSOL
 :
*end
```

NICE/SPAR

Linear Solution Procedures

Lockheed Palo Alto *CSM Testbed Development*

In this chart, an example of the simplest form of solution Procedure is shown: linear statics. Here, the ES Procedure is invoked to form all element stiffness matrices; Processor K is used to assemble the global stiffness matrix; INV is used to factor it; and SSOL is used to obtain a solution via forward/backward substitution. Subsequently, stresses may be computed, reactions checked, iterative refinement performed, etc.

NICE/SPAR

Nonlinear Solution Procedures

Lockheed Palo Alto *CSM Testbed Development*

```
*procedure NL_STATIC_1 ( beg_step = 1 ; beg_load = .1
                         max_step = 10 ; max_load = 1.0
                         max_iters = 7 ; max_cuts = 3
                         error_tol = 1.e-4 ; arc_scale = 1.0
                         nl_geom = 2 ; corotation = <true> )
  *do step = [beg_step], [max_step]
    *call STIFFNESS ( disp = <d_np1_i> )
    *do iter = 1, [max_iters]
      *call FORCE ( disp = <d_np1_i>; force = <r_np1_i> )
      *call SOLVE ( rhs = <r_np1_i> ; soln = <d_inc> )
      [xqt VEC
        <d_np1_ip1>  <-  <d_np1_i>  +  <d_inc>
        ROTATE <T_np1_i> * <d_inc>  ->  <T_np1_ip1>
```

NICE/SPAR

Nonlinear Solution Procedures

Lockheed Palo Alto *CSM Testbed Development*

This chart shows a very brief *excerpt* of a nonlinear static solution Procedure written for NICE/SPAR. The actual Procedure employs Crisfield's version of Rik's arc-length control algorithm, which is an adaptive load-stepping strategy for traversing complex curves in load/displacement space. At the core of the Procedure is the traditional iterative/incremental modified Newton-Raphson algorithm, with an arc-length constraint equation appended to determine the load increment as an additional unknown. Note that the Procedure argument list includes a number of strategy parameters (only a subset are shown) and involves DO loops, calls to other Procedures and direct Processor executions. In particular, notice the boxed section which illustrates one of many potential matrix-algebra oriented command languages. In this case the matrix language is indigenous to the VEC Processor, and is used to update the system displacement vector and nodal rotation triads at a given nonlinear iteration.

It is one of our goals to jointly develop a *standard/extendible* matrix language, and to make its interpretation an intrinsic part of the *architecture*. This would make procedure-writing more consistent, more natural and hence more straightforward.

Finally, note that no reference is made in Procedure `NL_STATIC_1` to the Generic Element *Procedure*: `ES`. Instead, the `ES` calls are covered with an even higher level of Procedures, such as `STIFFNESS` and `FORCE`, which form the element arrays (via `ES`) and assemble them into system arrays as well.

ORIGINAL PAGE IS
OF POOR QUALITY

NICE/SPAR PROCEDURES

ANALYSIS

Lockheed Palo Alto *CSM Testbed Development*

```
*procedure HINGED_CYLINDER ( elt_proc = ES1; elt_type = ANS9 ;
                             nel_x = 7 ; nel_y = 7 )

  *call GEN_CYLINDER ( radius = 100.; nel_x = [nel_x]; nel_y = [nel_y] ...)

  *call NL_STATIC_1 ( MAX_STEP = 20 )

*end
```

```
$ CSM
  *call HINGED_CYLINDER ( nel_y = 10; nel_x = 10 )
```

NICE/SPAR PROCEDURES

ANALYSIS

Lockheed Palo Alto *CSM Testbed Development*

The analyst with a problem to solve over and over again doesn't want to be bothered with the details of integrating pre-processing, solution and post-processing Procedures together each time. Hence, it is convenient to develop top-level Analysis Procedures for specific problems; with only those parameters that are volatile — i.e., those that the analyst is interested in changing — exposed. A trivial example of such an Analysis Procedure is shown in this view: a Procedure to analyze the classical hinged cylinder benchmark problem. Note that such Procedures are easy to write, and facilitate automatic parameter studies and/or standard quality assurance testing. A typical invocation of the Analysis Procedure is shown at the bottom of the chart, in which the command CSM is used to start-up the Testbed. Note that only the mesh parameters, `nel_x` and `nel_y`, are changed (from their default settings) by the call to Procedure `HINGED_CYLINDER`.

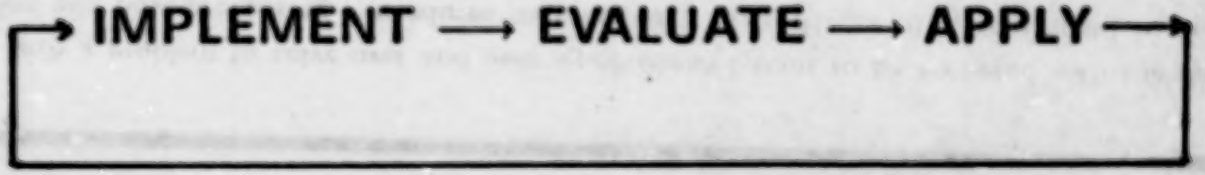
GENERIC ELEMENT PROCESSOR

Motivation

Lockheed Palo Alto *CSM Testbed Development*

- Adding new elements to the Testbed is difficult !!

- "Experimental Element" facility is inadequate — LINEAR only
- Testbed should provide common GAUNTLET (strainer)
- New elements should be easy to:



GENERIC ELEMENT PROCESSOR
MOTIVATION

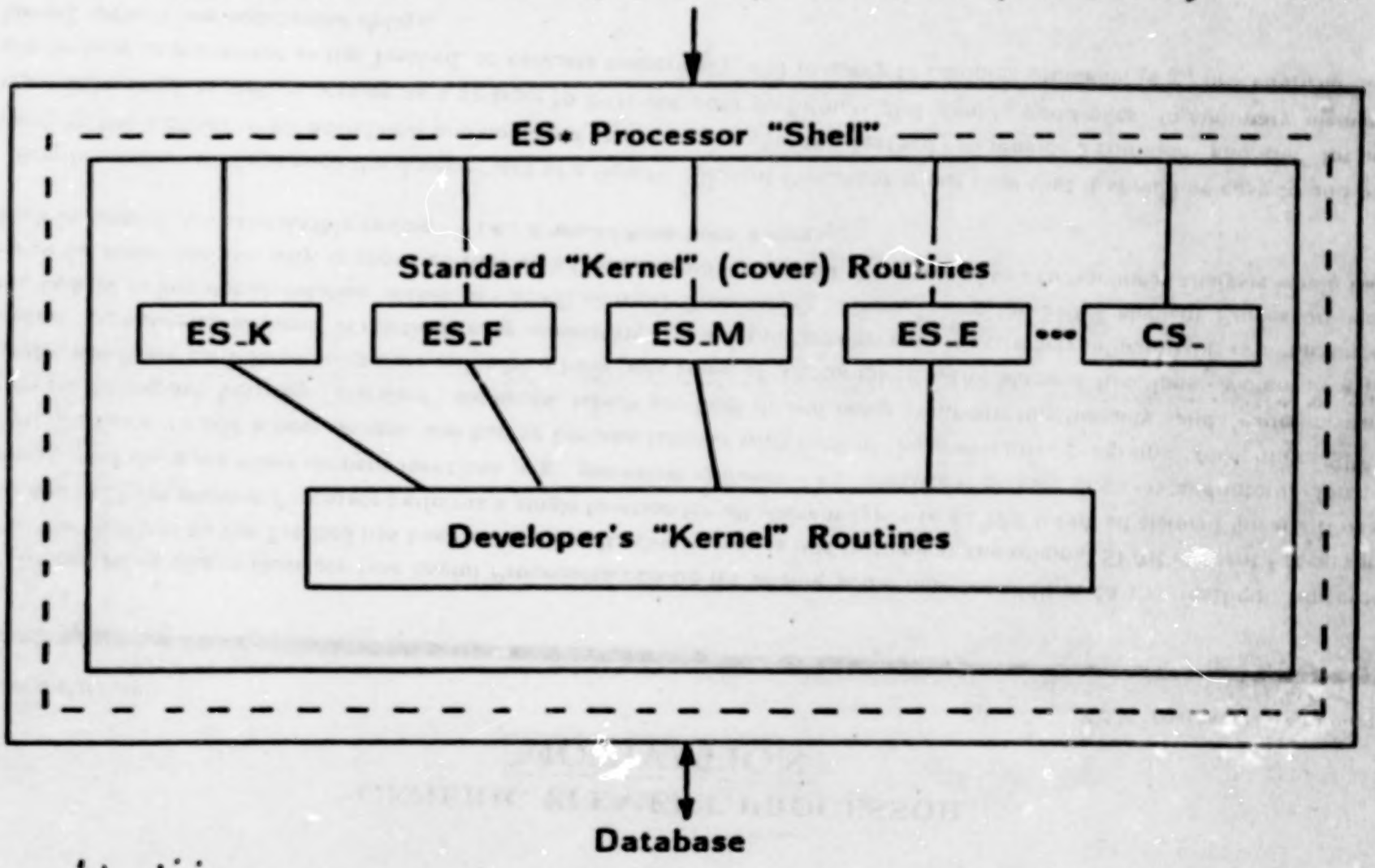
The preceding charts illustrate how useful Procedures can be for adding *some* new capabilities to the Testbed. However, adding new elements to the Testbed has been quite difficult. One reason is due to the way the original SPAR element Processors are organized. Each element Processor performs a single function for *all* element types (e.g., EKS forms all element linear stiffness matrices). And there are many element functions (e.g., geometric stiffness – KG, consistent mass – M, stress and internal force – GSF, etc.). Hence, to add a new element one had to become familiar with each of these executive programs. Also, these SPAR Processors distinguish between "standard" elements, which are built-in and enjoy complete functionality; and "experimental" elements, which are considered temporary and which have only some of the complete set of element functions. Moreover, while adding an "experimental element" is relatively easy, converting it to a "standard element" is extremely difficult (this is compounded by the lack of in-line documentation within the SPAR element Processors). Finally, since the SPAR element Processors were designed for linear analysis only, it appeared that an ad hoc attempt to extend their capabilities to nonlinear analysis would have resulted in tangled, unmaintainable coding — i.e., it would have been a kluge.

Another motive that spawned the development of a Generic Element Processor is our view that it should be easy to add new elements to the Testbed — for both linear and nonlinear analysis — so that the Testbed can provide a common "gauntlet" for new element technology, as well as serving as a strainer to filter out poor performers and identify good ones. In summary, elements should be easy to implement in the Testbed, to evaluate numerically, and to apply to complex problems (e.g., procedurized focal problems) without any substantial delays.

Generic Element Processor

APPROACH (visual)

Lockheed *CSM*



GENERIC ELEMENT PROCESSOR

APPROACH

Lockheed Palo Alto *CSM Testbed Development*

This chart gives a conceptual view of the Generic Element Processor. It features a standard outer software "shell" that processes user commands (such as `FORM STIFFNESS`) and handles all input/output from/to the global database (via GAL). Inside this shell is the non-standard "kernel" supplied by the element developer, which may be written in just about any style or granularity, provided that it is in FORTRAN (for now). Finally, a standard set of *shell/kernel* interface routines have been defined for each of the many element functions. The guts of these interface routines are provided by the element developer, while the subroutine names and argument lists are standardized. The function of the interface routines is to transition between the standard argument lists and the developers personal code in whatever manner is natural for the developer.

An additional function of the Processor "shell" is to provide automatic corotational updating capability for geometrically nonlinear (arbitrarily large rotation) analysis, so that element developers do not *have* to be concerned with such matters (however, they can override this option if they want to). Note that this automatic geometric nonlinearity feature has been implemented in a manner that is independent of the Testbed architecture (database and command language) — as an inner shell — to facilitate evolution of these two quite different functions.

GENERIC ELEMENT PROCESSOR

USER'S VIEW

Lockheed Palo Alto *CSM Testbed Development*

```
*call ES ( function = 'FORM { STIFFNESS | FORCE ...}'
           elt_dis_ds = TOT.DISP.<step>
           elt_rot_ds = TOT.ROTN.<step> ...)
```

```
*procedure ES ( function; elt_dis_ds; elt_rotn_ds ...)

    *do es_proc = 1, num_es_proc

      [xqt <es_proc_name>
       [function]

    *enddo

*end
```

GENERIC ELEMENT PROCESSOR

USER'S VIEW

Lockheed Palo Alto *CSM Testbed Development*

The Generic Element Processor is not really a single Processor at all, but rather a standard Processor "shell" that can be used to construct an unlimited number of specific element Processors. The user (i.e., analyst or Procedure writer) interface to element Processors created from the Generic Element Processor mold can be either direct — through commands such as `FORM STIFFNESS`, `FORM FORCE`, etc., or through a high-level Procedure called `ES` (for *S*tructural *E*lements). The advantage of the Procedure interface is that it will run all element Processors that are active for the current problem, ***automatically***, via database lookups and internal `DO` loops. This saves the user the trouble of explicitly executing each element Processor for each function required. Note that the `ES` Procedure call has arguments that are analogous to the Processor commands. For example, `function = 'FORM STIFFNESS'`, defines the command to be executed by all structural element Processors, and `elt_dis_da = TOT.DISP.<step>`, defines the name of the displacement dataset to be used by all structural element Processors.

GENERIC ELEMENT PROCESSOR

DEVELOPER'S VIEW

SHELL/KERNEL INTERFACE ROUTINES

ES_D	Parameter Definition
ES_I	Initialization
ES_KM	Material Stiffness
ES_KG	Geometric Stiffness
ES_FI	Internal Force
ES_FE	External Force
ES_MC	Consistent Mass
ES_MD	Diagonal Mass
ES_TR	Transformations
ES_N	Unit Normal Vectors

GENERIC ELEMENT PROCESSOR
DEVELOPER'S VIEW

Lockheed Palo Alto *CSM Testbed Development*

For the element developer, of main interest is the set of shell/kernel interface routines that must be completed to bring a newly implemented element up to full functionality. A partial list of these routines is shown in this chart. (The list is subject to grow as research demands.)

While most of these routines correspond to Processor commands (which the element developer doesn't have to deal with), more than one shell/kernel interface routine may be invoked as a result of a single command. For example, the command `FORM FORCE/INT` may cause `ES_D` (element definition), `ES_E` (element strains) and `ES_FI` (element internal force vector) to be called by the generic Processor shell. Note also that the element developer does not currently have to provide constitutive routines. A standard set of *linear* constitutive routines is built in to the shell. Eventually, we expect this standard constitutive interface to evolve into a separate Generic Constitutive Processor for general linear and nonlinear material models (including stress/tangent-modulus calculation and failure analysis).

GENERIC ELEMENT PROCESSOR

DEVELOPER'S VIEW (continued

ES_D (eltnam, eltnum, ctls, DEFS, DOFS, NODES, PARS, Status)

ES_FI (eltnum, ctls, defs, dofs, nodes, pars, store, x, d, s, FI, Status)

DEFS(pdNEN) — number of element nodes
(pdNIP) — number of integration points
(pdCLAS) — idBEAM | idSHEL | idSOLI
(pdNDOF) — number of freedoms per node

(pdESKM) — ES_KM implementation status

:

DEVELOPER's VIEW (continued)

This chart provides a closer look at the shell/kernel interface routines, showing sample argument lists and some of the important parameters contained in one of the argument arrays. Notice that we are compensating for FORTRAN as much as possible, by using *array* data structures (wherever appropriate) to facilitate later extensions.

Subroutine ES_D (element Definition) is different from the other interface routines. For it, the element name (eltnam) is input, and the basic element definition parameters (DEFS), a nodal freedom activity table (DOFS) and a node sequence array (NODES) are output. In contrast, action routines such as ES_FI (element internal force) employ DEFS, DOFS and NODES as input — in addition to a control array (CTLS) and a research parameters array (PARS). In this case, the element nodal coordinates (x), displacements (d) and stresses (s) are input, and the internal force vector (FI) and subroutine return status (Status) are output.

The DEFS array is an expanding list of parameters that enable the element developer to convey a precise description of the element to the generic Processor shell — for example to enable the shell to perform certain corotational functions. It also contains parameters such as ESKM, ESKG, ESFI, etc., which convey the implementation status of the corresponding element function. The entire DEFS array is stored in the global database for users (and developers) to query before, during or after an analysis.

GENERIC ELEMENT PROCESSOR

FEATURES

Lockheed Palo Alto *CSM Testbed Development*

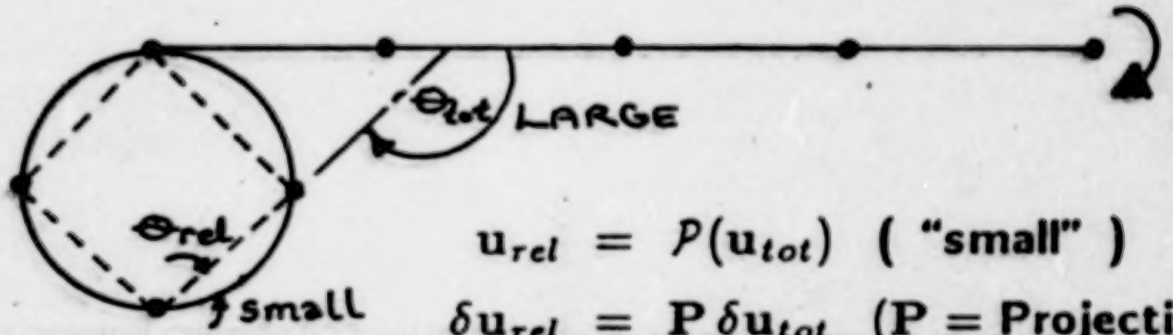
- **MODULARITY**
  - — Developers may work independently (insulated from arch.)
  - — All elt. fns under one roof (object-oriented)
- **EXTENDIBILITY**
  - — Number of elt. types/fns/parameters open-ended
  - — Number of elt. Processors open-ended (ES1, ES2, ...)
- **SIMPLICITY**
  - — Shell/kernel interface easy to build (repackage in days!)
  - — Procedure interface is high-level (e.g., FORM STIFFNESS)
- **SELF-DESCRIPTION**
  - — Standard interface makes elt. software accessible to others
  - — Database contains elt. parameters/status for user query
- **AUTOMATIC GEOMETRIC NONLINEARITY (Corotational)**

GENERIC ELEMENT PROCESSOR
FEATURES

Lockheed Palo Alto — *CSM Testbed Development*

In summary, the Generic Element Processor is modular, extendible, simple to use/develop, self-described and facilitates geometric nonlinearity. This last feature is made possible by implementation of a corotational utility library within the Processor shell. This corotational shell enables linear elements to be employed for problems involving arbitrarily large rotations — with no additional work by the developer beyond the usual linear subroutines (including one for the geometric stiffness matrix). Moreover, while the corotational option is the default, alternate methods for handling geometric nonlinearity are accommodated by providing the developer with the element displacement vector as input argument to most of the shell/kernel interface routines. Also, the special nodal rotation quantities that are automatically provided in large rotation analysis for beam and shell elements (i.e., rotation pseudovectors corresponding to orthogonal triads) can either be employed, ignored, or replaced, by the element developer who wishes to incorporate a completely different type of nonlinear formulation. Note that total Lagrange (TL), updated Lagrange (UL) and corotated Lagrange (CL) formulations all fall naturally within the present framework.

COROTATION
(In a Nutshell)



$$\mathbf{u}_{rel} = P(\mathbf{u}_{tot}) \quad (\text{"small"})$$

$$\delta\mathbf{u}_{rel} = \mathbf{P}\,\delta\mathbf{u}_{tot} \quad (\mathbf{P} = \text{Projection})$$

$$\mathbf{K}_{tot} = \mathbf{P}^T\mathbf{K}_{rel}\mathbf{P} + \mathbf{K}_{hot}$$

— Upgrades linear elements to geom. nonlinear (ELEMENT-INDEPENDENT)

— Projects out rigid-body errors (warping sensitivity)

— Adds h.o.t.s for Consistent Tangent Stiffness [Rankin/Nour-Omid]

— Uses consistent NODAL rotation update: $\mathbf{R}_{n+1} = e^{\delta\theta}\mathbf{R}_n$

COROTATION
(In a Nutshell)

Lockheed Palo Alto

CSM Testbed Development

Corotation is a method for making large rotations relative to an inertial frame *look* like small rotations at the element level. This is achieved by defining an element coordinate frame (for each element) that rotates with the element (as defined by its nodal coordinates). The rigid motion of this frame is then "subtracted" from the total motion of the nodes, leaving relative translations and rotations that can be made arbitrarily small by simply refining the mesh (i.e., by adding more element frames). Note that for the rotational degrees of freedom, "subtracting" the rigid rotations really amounts to multiplication of orthogonal matrices, and that the updating of these matrices (or triads) at nodes from step to step (before removing the rigid motion) must be done in a manner that preserves orthogonality.

Once the relative motions at nodes have been rendered sufficiently small, they may be used in simplified strain-displacement relations. For example, for shell elements, either linear strain-displacement relations or so-called moderate rotation nonlinear strain-displacement relations may be used in conjunction with the corotational procedure. The effect of using nonlinear versus linear element strain-displacement relations usually amounts to an increase in accuracy, which can alternatively be achieved by adding more elements.

Corotation is not a new idea (e.g., Belytchko [5], Wempner [6] and others were some of the innovators) but we have given it some new twists [7]. First, we have made it more *element-independent* by generalizing it to beam, shell and solid elements and providing generic software utilities for all three classes; and second, we have established a firm mathematical foundation by deriving it from variational principles and identifying its correspondence to a *projection* operator (P). By capitalizing on this projection operator idea, we have — through consistent linearization of the variational functional — come up with an element-independent higher-order stiffness matrix that ensures quadratic convergence in the context of a Newton-Raphson nonlinear solution procedure. A suprising fringe benefit is that the projection matrix, $\mathbf{P}$, that emerges from linearization, seems to correct rigid body errors (e.g., warping sensitivy) even for linear analysis! This may have a profound affect on various old and new element formulations (e.g., Szabo's p-refinement elements have implicit rigid body errors for low-order *polynomial* approximations).

GENERIC ELEMENT PROCESSOR

PRIOR STATUS (August)

Lockheed Palo Alto *CSM Testbed Development*

- **Operational for Linear and Geom. Nonlinear Shell Statics**
- **Element Processors:**
  - — ES1 C^0 **shell elements (4/9-ANS and 4/9/16-SRI)**
  - — ES2 C^1 **shell elements (4/"9"-HYB)**
- **Analysis Procedures:**
  - — **L_STATIC_1**
  - — **L_STABIL_0 / L_STABIL_1**
  - — **NL_STATIC_1** [Crisfield/Riks +]
- **Applications**
  - — **"Patch" Tests**
  - — **Stiffened Shell Buckling**
  - — **Simple Shell Post-Buckling**

GENERIC ELEMENT PROCESSOR

PRIOR STATUS (August)

Lockheed Palo Alto *CSM Testbed Development*

As of last August, we had used the generic element Processor "shell" to construct two specific element Processors, namely, ES1, which contains a family of C^0 shell elements, and ES2, which contains a family of C^1 shell elements. The C^0 (transverse shear deformable) shell elements in ES1 include 4- and 9-node assumed natural-coordinate strain (ANS) elements by Park and Stanley [8], and various selective/reduced integrated (SRI) elements (as in [9]). The C^1 (transverse shear rigid) shell elements in ES2 include flat and curved (exact-geometry) 4-node hybrid (HYB) elements by Kang and Pian [10]. Both the ANS and HYB elements in the two Processors, respectively, are formulated in terms of a local natural-coordinate basis, employing covariant strain components. Hence mesh distortion sensitivity is reduced considerably with respect to elements that employ local-Cartesian strain components for discretization. Collectively, Processors ES1 and ES2 represent (we believe) the state-of-the art in shell element technology.

We had also developed Procedures for linear and nonlinear static analysis, and had applied Processors ES1 and ES2 (via these Procedures) to simple linear and nonlinear test cases.

GENERIC ELEMENT PROCESSOR

Δ-PROGRESS (Aug–Nov)

Lockheed Palo Aito *CSM Testbed Development*

- Cleanup/Testing (Iteration)
- New Element Processors:
  - — ES3 Modified SPAR Hybrid Solid elts [Aminpour]
  - — ES4 " " " Shell " "
  - — ES5 STAGS 410 Shell elt [Brogan]–NAVY
  - — ES6 2/3-Node Hybrid Beam (Truss/Stiffener) [Kang]
  - — ES6 Improved ANS Shell [Park/Stanley/Cabiness] –NAVY
- New Analysis Procedures:
  - — NL_STATIC_1 extended to Specified Displacements
- New Applications:
  - — Postbuckling of Composite Stiffened Panels

GENERIC ELEMENT PROCESSOR

Δ-PROGRESS (Aug–Nov)

Lockheed Palo Alto

CSM Testbed Development

Between August and November, we have performed several iterations on the generic element Processor "shell", making it easier for both element developers and Procedure writers to interface with, and adding some new pre- and post-processing capabilities as well (e.g., stress transformation/archival and automatic freedom supression commands).

New element Processors, constructed with the generic element Processor shell, were implemented both at LPARL and at LARC. Dr. Mohammad Aminpour (LARC) re-packaged a set of improved SPAR hybrid shell and solid elements within Processors ES4 and ES3, respectively, following a one-week visit by Dr. Aminpour to LPARL. Subsequently, we re-packaged the workhorse shell element used in the STAGS code [11] (therein called the "410" element) as Processor ES5.* We also made a preliminary implementation of a set of straight and curved C^1 hybrid beam elements within Processor ES6; these elements are compatible with the shell elements implemented in Processor ES2 by Dr. David Kang.

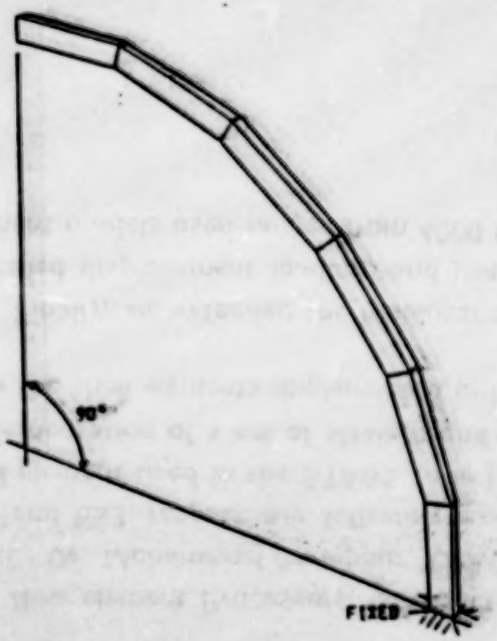
Finally, we extended the nonlinear static analysis Procedure, `NL_STATIC_1`, based on Crisfield's arclength method [12], to specified displacement loading, and used it to analyze the postbuckling response of I-stiffened composite panels. The finite element models used ranged from 4000 to 9000 degrees of freedom.

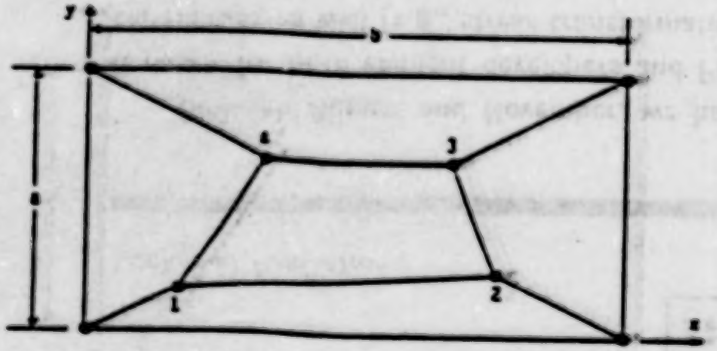
---

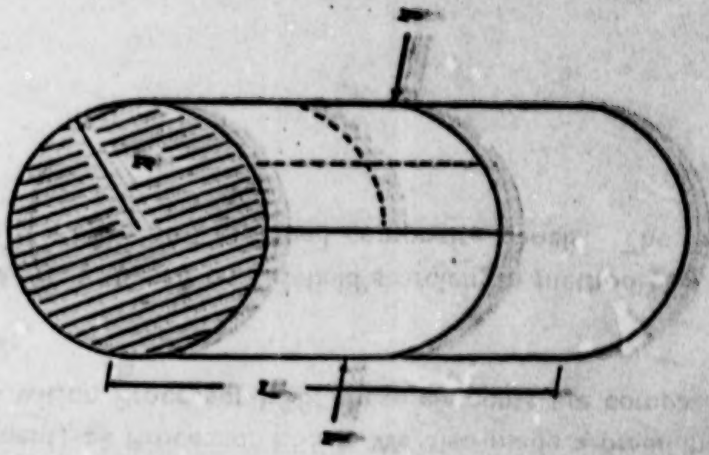
* The NAVY (NSRDC) has also contributed some funding towards this effort.

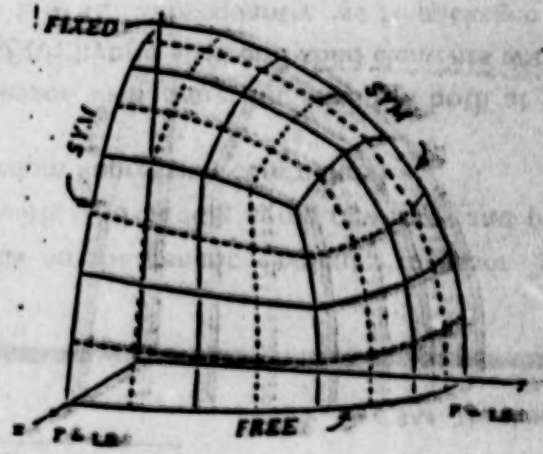
ELEMENT TEST CASES

LINEAR







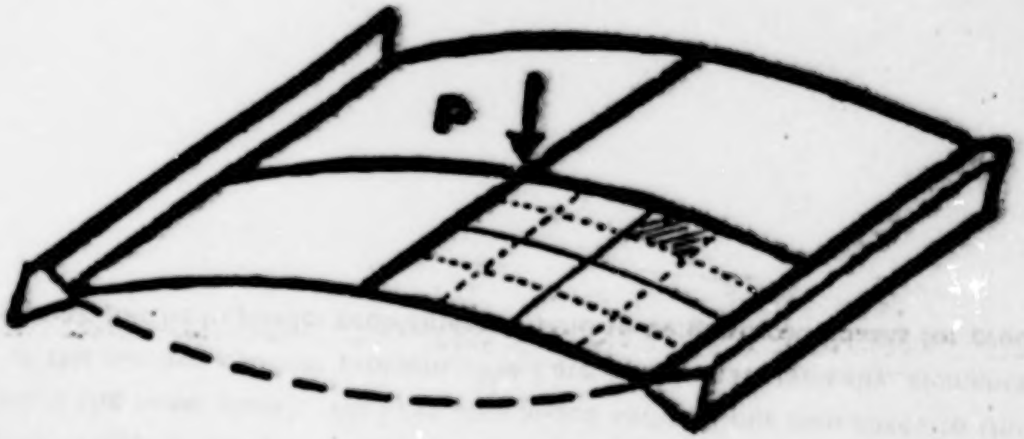


ELEMENT TEST CASES
LINEAR

Before applying a new element or solution algorithm to a complex problem, it is wise to first verify/evaluate it using standard benchmark problems. Shown in this chart are some of the linear test cases proposed by MacNeal/Harder [13]. They represent a series of patch tests and covergence tests that tend to bring out element pathologies. For example, the pinched hemisphere (bottom right) requires shell elements to display nearly inextensional bending behavior in locally warped configurations (quadrilateral element corner points do not all lie in the same plane). We have also added some of our own cases to this gauntlet, including gradually distorted mesh versions of the pinched cylinder problem. We have found that, typically, elements that can handle the pinched hemisphere and distorted-mesh pinched cylinder problems well tend to be good candidates for production applications.

ELEMENT TEST CASES

NONLINEAR: Hinged Cylinder



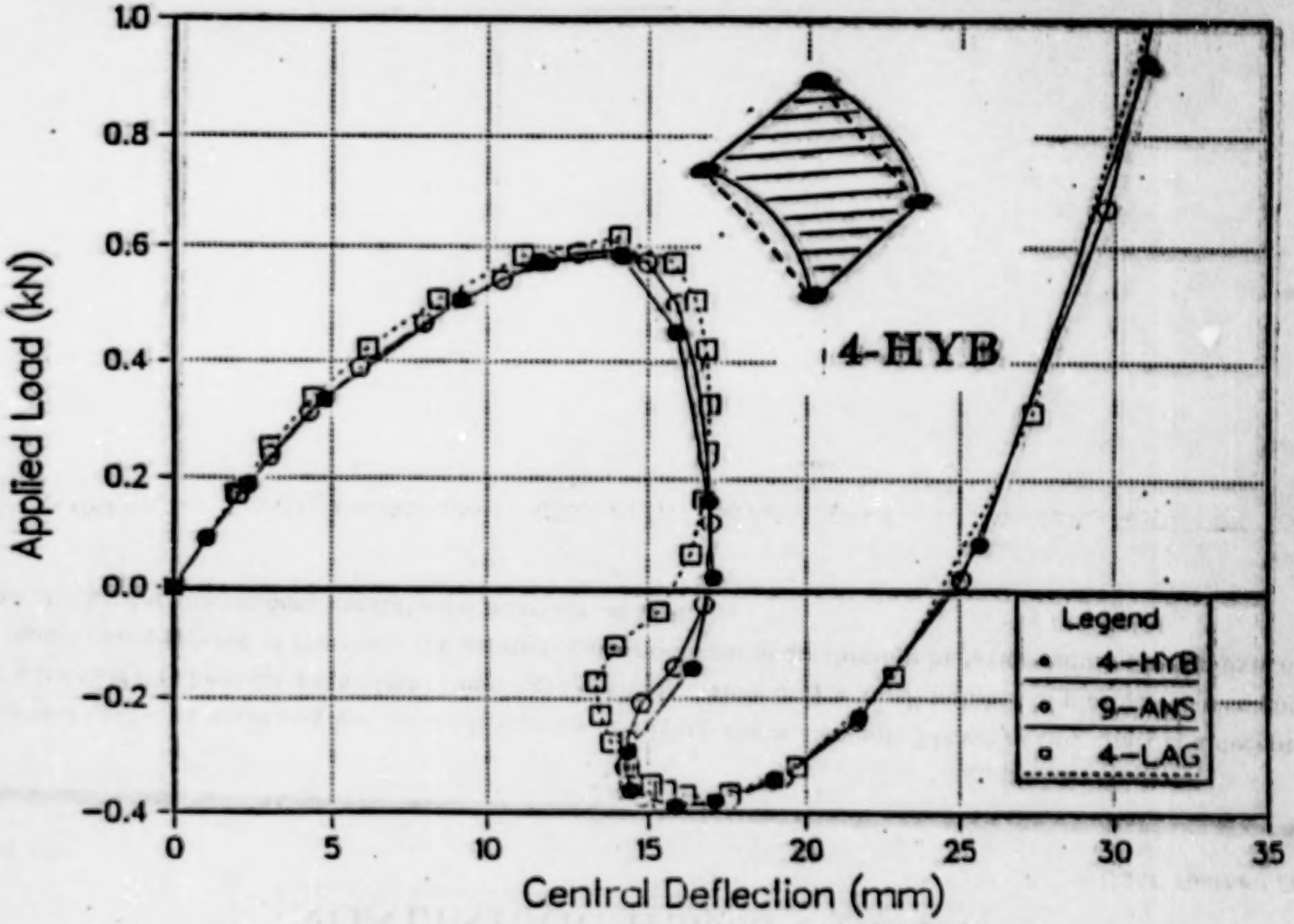
NONLINEAR: Hinged Cylinder

The linear test cases are necessary but not sufficient for certifying a new element. Shown in this chart is a deceptively simple nonlinear test case that has become a standard: the hinged cylinder. While only a small number of plate/shell elements should be necessary to model one quadrant of the shell, the geometrically nonlinear snap-through behavior requires a sophisticated solution algorithm, and checks out the element corotational interface as well.

HINGED CYLINDER RESULTS

C^0 vs C^1 Elements

Lockheed Palo Alto *CSM Testbed Development*



HINGED CYLINDER RESULTS

C^0 vs C^1 Elements

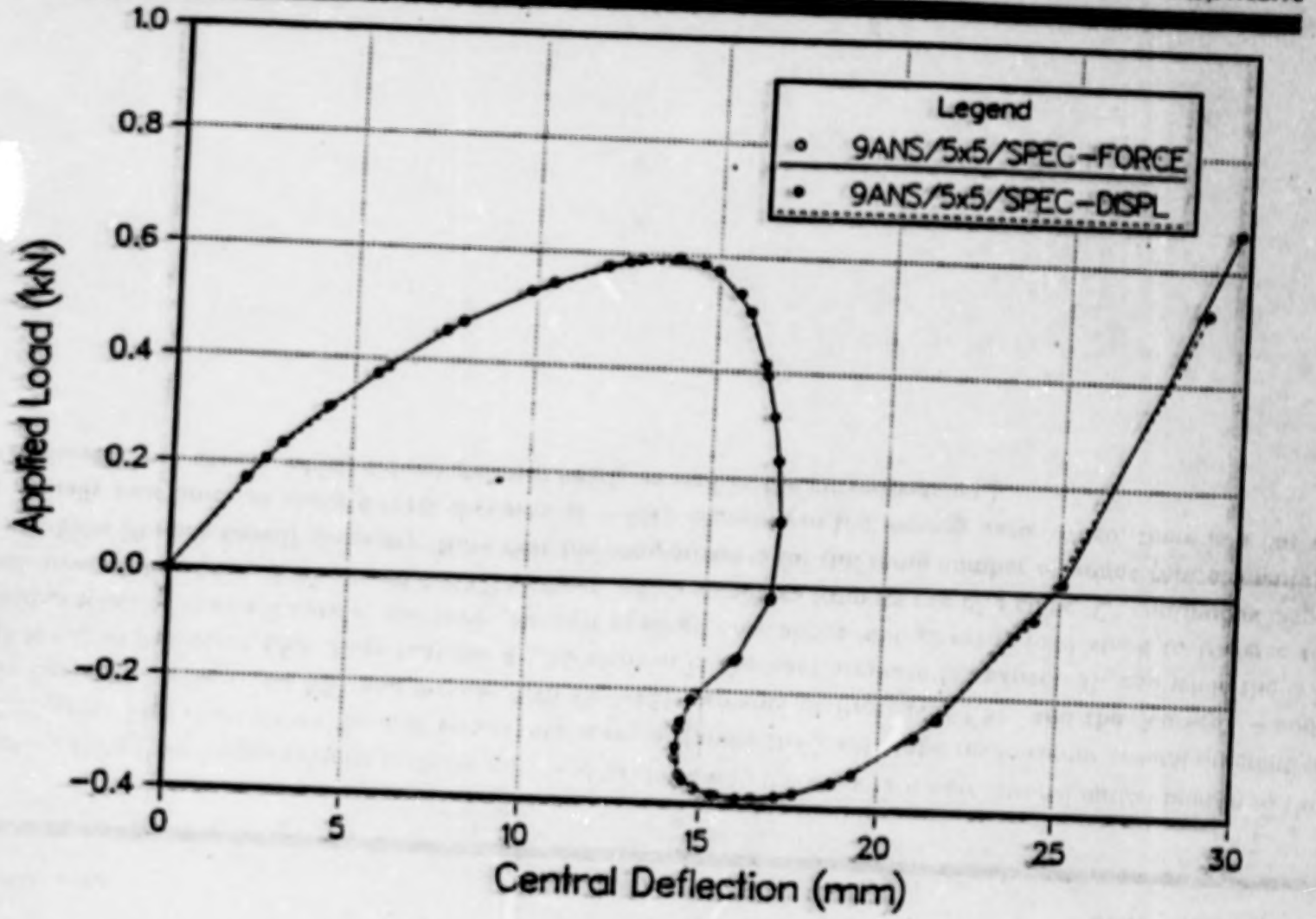
Lockheed Palo Alto *CSM Testbed Development*

One of the things the hinged cylinder problem was used to check was the automatic corotational option built in to the generic element Processor. This chart shows the load versus displacement (under the load) static response for several different elements: the 4-node Lagrange element (4-LAG) and 9-node ANS (9-ANS) elements of Processor ES1, and the "curved" 4-node hybrid element (4-HYB) of Processor ES2. Note that the 4-LAG element is the least accurate (as expected), and while the 9-ANS and 4-HYB yield practically identical curves, the latter element required only about half as many load steps to traverse the curve. This is indicative of the power of the curved 4-HYB element, which emanates from its use of a cubic, C^1 continuous displacement field and parabolic (9-node based) geometry. Note that the comparison is for the same number of *nodes* (not elements), so that there are actually four times as many 4-HYB elements as 9-ANS elements in the meshes used. (Also, there is a flat version of the 4-HYB element, not shown, which did not perform nearly as well as the curved version.)

HINGED CYLINDER RESULTS

Specified Disp. vs Force

Lockheed Palo Alto — *CSM Testbed Development*



HINGED CYLINDER RESULTS

Specified Disp. vs Force

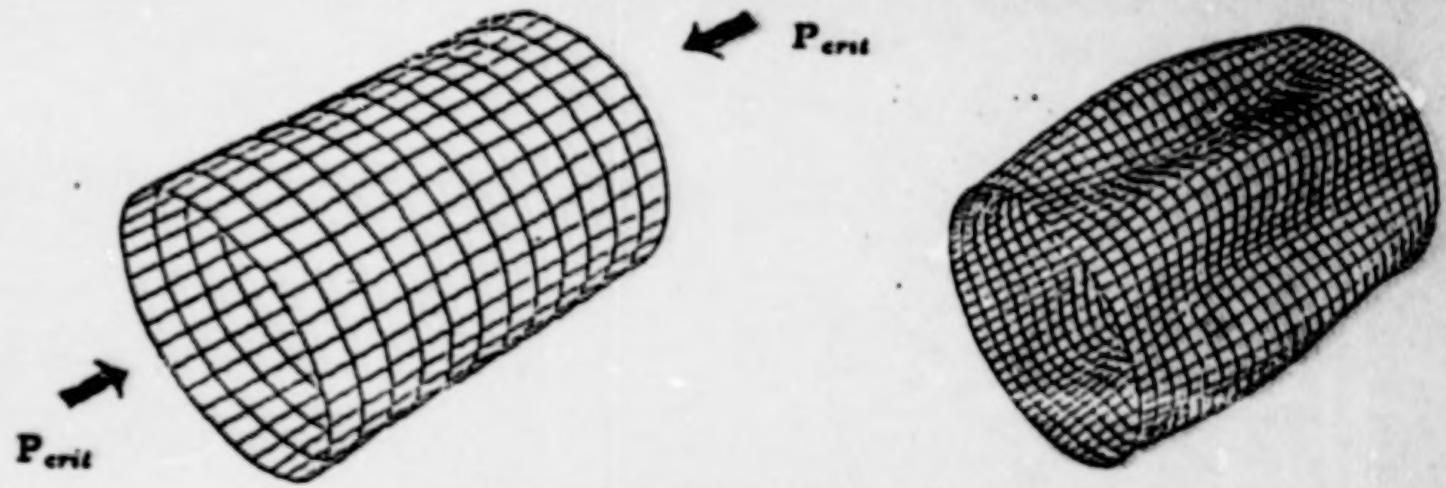
Lockheed Palo Alto *CSM Testbed Development*

The second thing checked by the hinged cylinder problem was the newly added specified displacement loading capability to Procedure NL STATIC.1. Shown in this chart are the central force vs displacement curves for both specified force loading and specified displacement loading. Note that there is relatively little difference between the curves or the number of load steps required. However, the load steps are different, due to differences in the scaling of the arclength constraint equation and in the error norms used for the two cases.

ELEMENT TEST CASES

NONLINEAR: Compressed Cylinder

Lockheed Palo Alto *CSM Testbed Development*



CLASSICAL BUCKLING PROBLEM

(Same Eigenvalue: $\lambda = P_{crit}$)

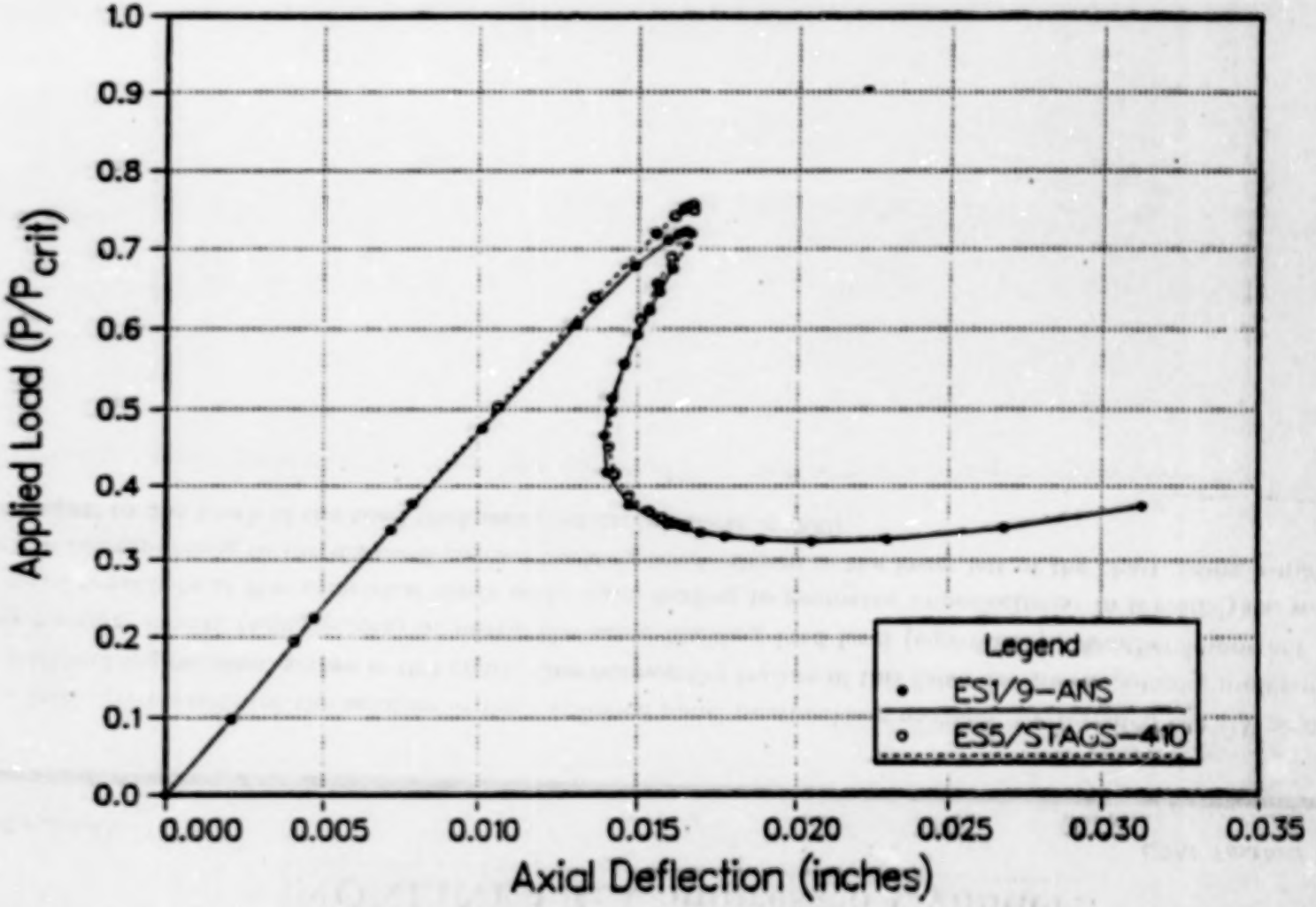
NONLINEAR: Compressed Cylinder

As a final pre-requisite for the solution of the I-stiffened panel postbuckling problem, we revisited the classical compressed cylinder postbuckling problem, shown in this chart. One noteworthy feature of this (also deceptively simple) problem is that there are many buckling modes (eigenvectors) at nearly the same buckling load level (eigenvalue). Another important, and related, feature is the sensitivity of thin cylindrical shells under axial loading to geometric imperfections. In the analysis, we imposed an imperfection corresponding to the diamond-pattern buckling mode shown in the lower left of the chart, using a maximum radial amplitude equal to one-tenth of the shell thickness (radius/thickness = 300).

COMPRESSED CYLINDER RESULTS

9ANS vs E410

Lockheed Palo Alto *CSM Testbed Development*



COMPRESSED CYLINDER RESULTS

9-ANS vs E410

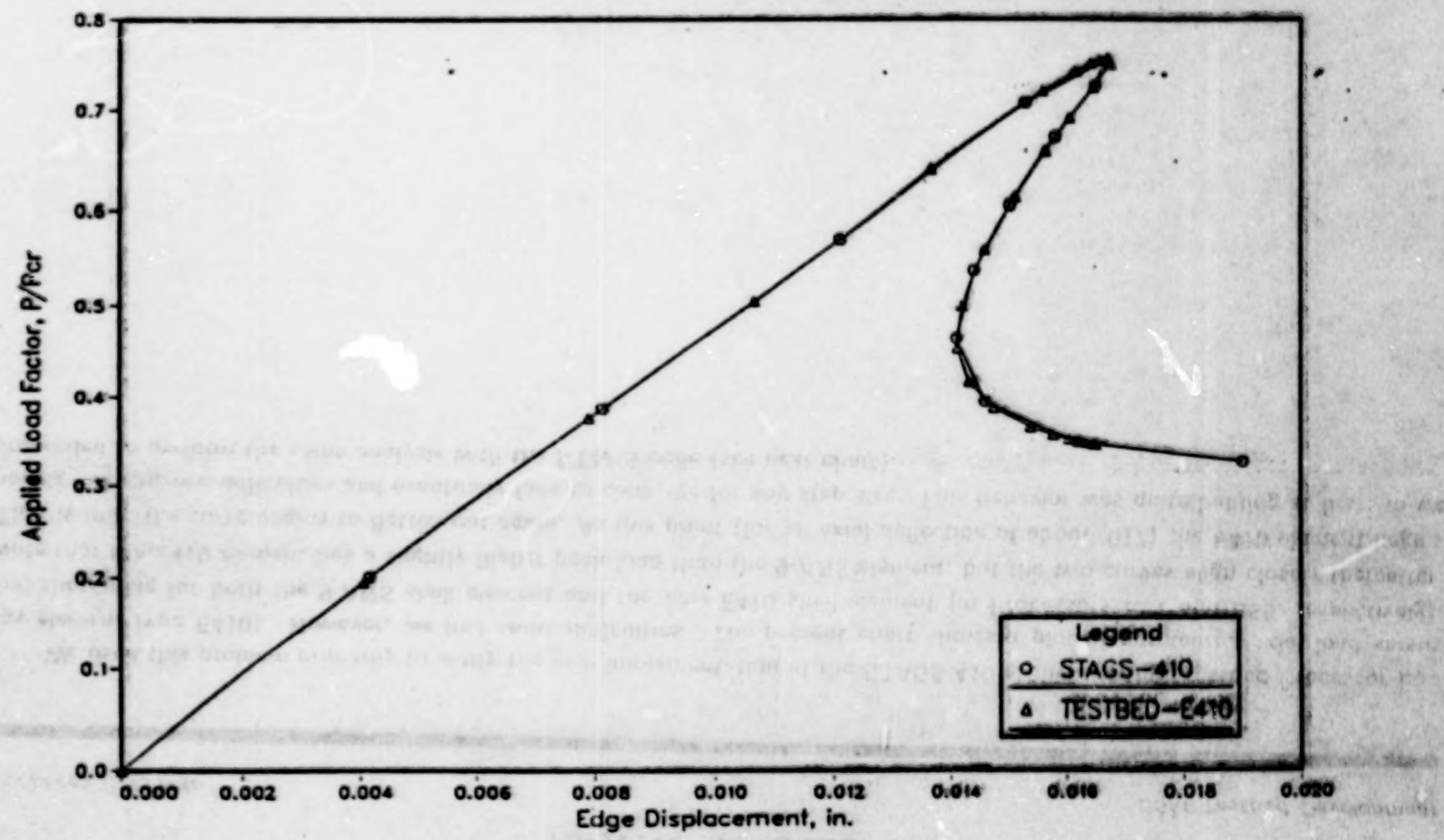
Lockheed Palo Alto *CSM Testbed Development*

We used this problem primarily to verify the new implementation of the STAGS 410 element within Testbed Processor ES5 (as element type E410). However, we had some difficulties. The present chart shows a plot of normalized axial load versus end-shortening for both the 9-ANS shell element and the new E410 shell element (in Processors ES1 and ES5, respectively). Note that the E410 element has a slightly higher peak load than the 9-ANS element, but the two curves align closely thereafter. That is until the curve begins to flatten out again. At this point (for an axial deflection of about .017) the E410 element begins having convergence difficulties and eventually fails to converge for any step size. This behavior was quite baffling at first, so we proceeded to perform the same analysis with the STAGS code (see next chart).

COMPRESSED CYLINDER RESULTS

STAGS vs CSM (E410)

Lockheed Palo Alto *CSM Testbed Development*



630

COMPRESSED CYLINDER RESULTS

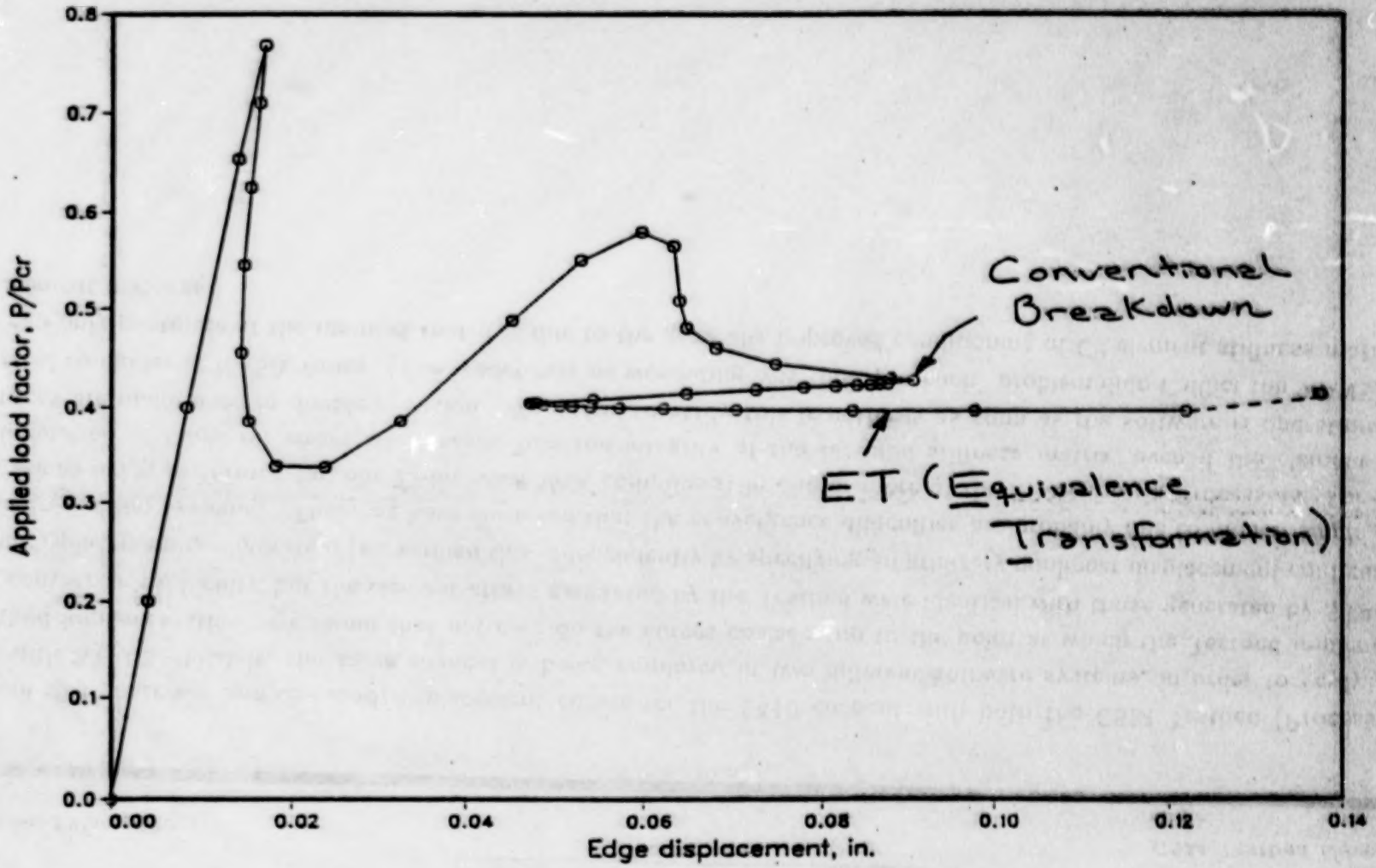
STAGS vs CSM (E410)

In this chart we compare load/displacement curves for the E410 element with both the CSM Testbed (Processor ES5) and with STAGS. That is, the same element is being employed in two different software systems, in order to verify the new Testbed implementation. We found that not only do the curves coalesce up to the point at which the Testbed implementation has convergence difficulty, but the element arrays generated by the Testbed were identical with those generated by STAGS for a given displacement configuration (we verified this independently by specifying an arbitrary nonlinear displacement configuration as input to the two systems). Thus, we have surmised that the convergence difficulties are probably due to hidden *single precision* operations being performed (on our 32-bit word VAX computers) in one or more of the SPAR matrix Processors. For example, accumulation of round-off errors could easily ruin the integrity of the factored stiffness matrix, even if the element stiffness matrices are maintained in double precision. We intend to verify this hypothesis as soon as the software is operational on the CRAY-2 computer at NASA/Ames. (The reader may be wondering why this "precision" problem didn't afflict the 9-ANS element. We can only postulate at the moment that it is due to the generally improved conditioning of C^0 element stiffness matrices over C^1 element matrices.)

COMPRESSED CYLINDER RESULTS

STAGS ET Breakthrough

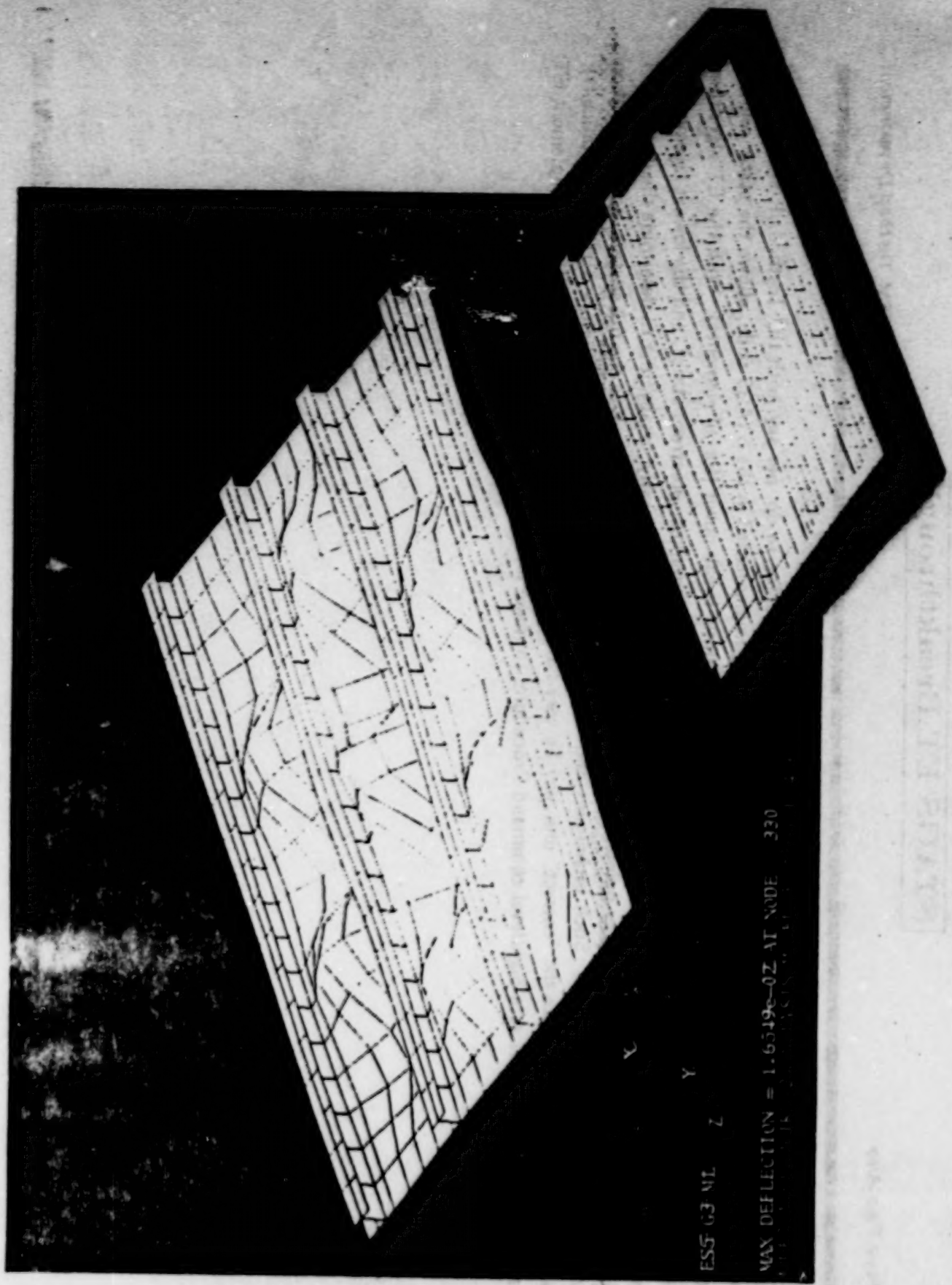
Lockheed Palo Alto *CSM Testbed Development*



COMPRESSED CYLINDER RESULTS

STAGS ET Breakthrough

The last curve to be shown here for the compressed cylinder has nothing to do, per say, with the Testbed. It shows how a new algorithm, called ET (Equivalence Transformation) [14], recently implemented in the STAGS code, allows the static solution to proceed beyond the point at which conventional, arclength algorithms break down (for all elements). With the ET algorithm, eigenvalue analyses are performed about these barrier points (i.e., using a nonlinear prestress state), and the lowest eigenvectors are used to temporarily reduce the system. The reduced system is then solved nonlinearly, locally, just to determine the direction of a stable equilibrium path beyond the barrier point. The reduced system is then expanded back to the full system, and the solution continued on the newly found path. Such techniques allow the user to explore solutions never before attainable (statically): perhaps too many solutions, since the "correct" one may be very hard to deduce. Nevertheless, we have plans to implement ET within the Testbed next year, as a high-level command Procedure, to make it more accessible to others.



ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

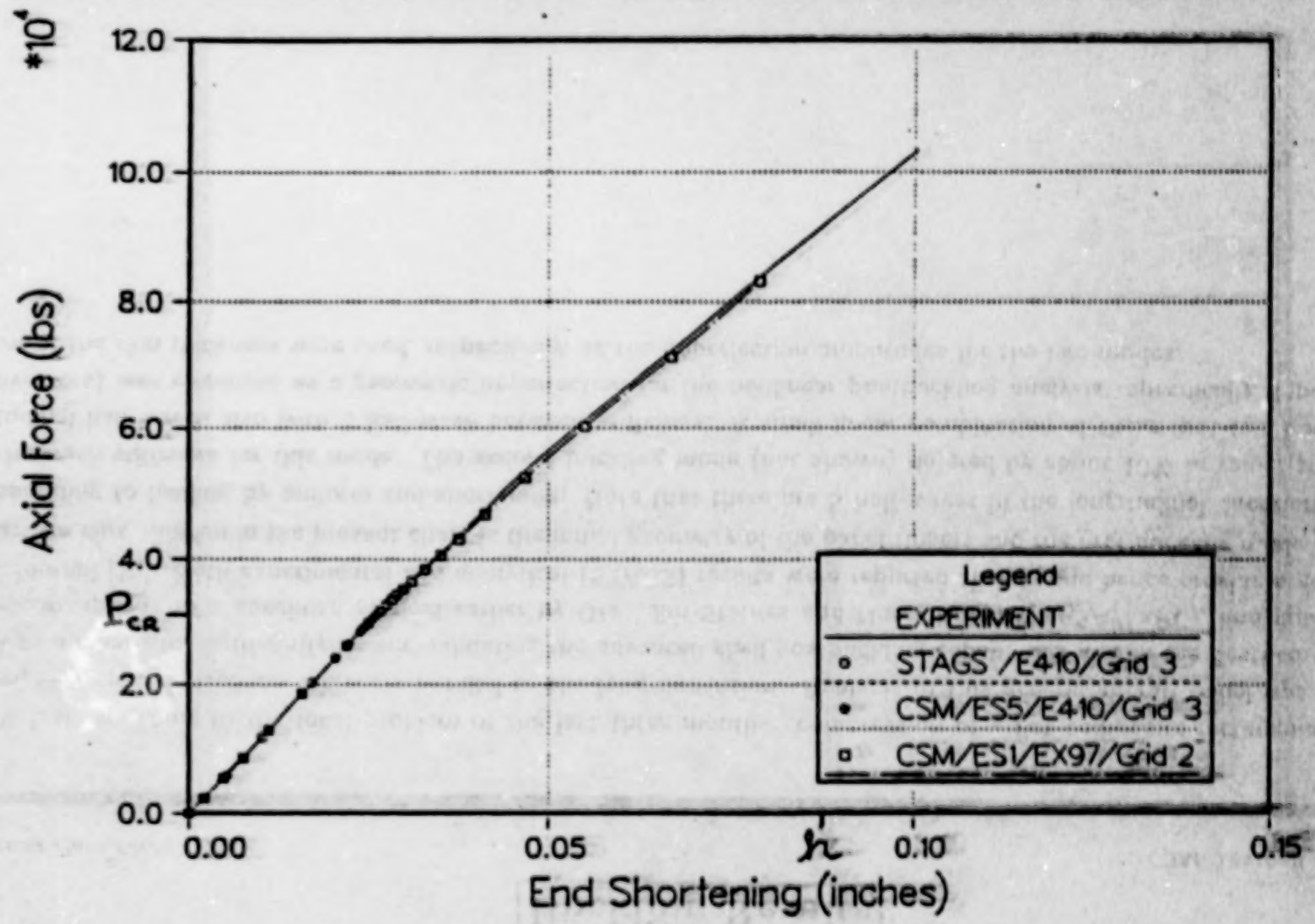
I-STIFFENED PANEL
Buckling Results

At last, we come to the focal problem of the last three months: compression of a flat composite rectangular panel with four, equally spaced I-section stiffeners parallel to the long dimension. Analysis of this generic aircraft panel was proposed by NASA as a means for both verifying and validating the advanced shell postbuckling capabilities now in the Testbed. The specific choice corresponds to a specimen studied earlier by Drs. Jim Starnes and Norm Knight (NASA/LaRC), and published in the AIAA Journal [15]. Both experimental and analytical (STAGS) results were reported therein and hence provide a good basis for comparison here. Shown in the present chart is the initial geometry of the panel (inset) and the first buckling mode (eigenvector) corresponding to loading by uniform end-shortening. Note that there are 5 half-waves in the longitudinal direction, and 1 half-wave between stiffeners for this mode. The second buckling mode (not shown) differed by about 10% in value, and featured 6 longitudinal half-waves also with 1 half-wave between stiffeners. A small linear combination of these first two buckling modes (eigenvectors) was employed as a geometric imperfection for the nonlinear postbuckling analysis: specifically, 1 percent and .1 percent of the skin thickness were used, respectively, as the imperfection amplitudes for the two modes.

I-STIFFENED PANEL RESULTS

Load vs End-Shortening

Lockheed Palo Alto *CSM Testbed Development*



I-STIFFENED PANEL

Load vs End-Shortening

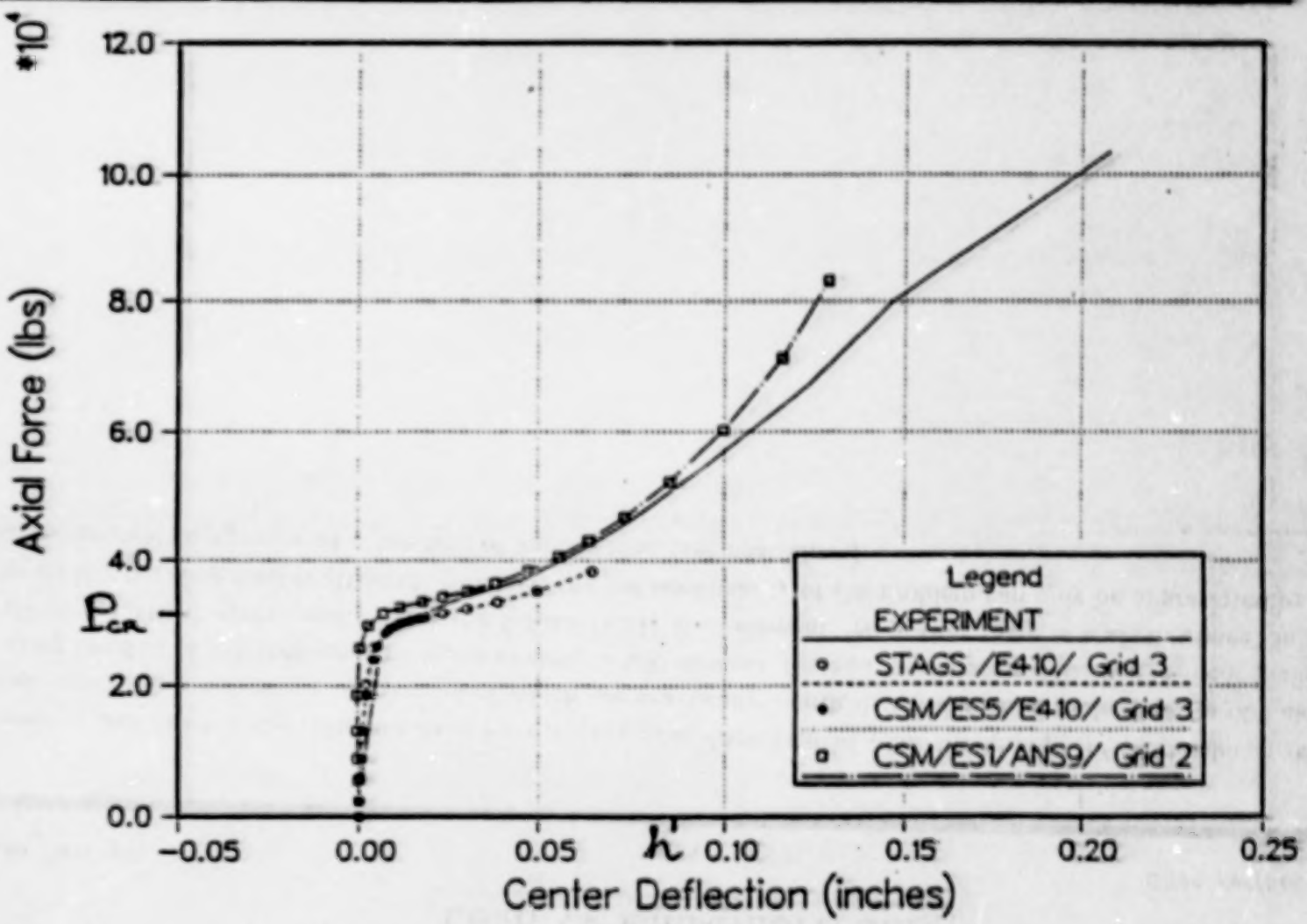
Lockheed Palo Alto *CSM Testbed Development*

Shown in this chart is the resultant axial force versus axial displacement (end-shortening) for the nonlinear response of the I-stiffened panel. Four curves are plotted, all of which are practically identical: experimental (solid line), a 600 element STAGS model using the 410 (ES5) shell element (open circles), a 600 element Testbed model using 410 shell element (solid circles), and a 400 element Testbed model model using the 9-ANS (ES1) shell element. Note that there is a slight "knee" in the curve just after the critical buckling load is attained. However, the full nonlinearity of the problem can only be appreciated by observing the lateral displacement component as a function of applied load (see next chart).

I-STIFFENED PANEL RESULTS

Load vs Center Deflection

Lockheed Palo Alto *CSM Testbed Development*



I-STIFFENED PANEL

Load vs Center-Deflection

Lockheed Palo Alto *CSM Testbed Development*

In this chart the lateral displacement at the center of the I-stiffened panel is plotted versus the applied axial force, for the same cases as shown in the previous chart. Here, we can see the pronounced nonlinearity (i.e., "knee") as the load is increased beyond the critical buckling value. We can also see that, as before (see the Compressed Cylinder results), while the STAGS solution follows the experiment all the way to collapse, the same element in the Testbed (ES5/E410) has unresolvable convergence difficulties as it rounds the bend. Once again, we tentatively attribute this phenomenon to hidden single precision matrix operations.

Notice also that while the Testbed solution for the 9-ANS element (ES1/ANS9) does not have convergence difficulties, it begins to diverge from the experimental curve at about half of the collapse load. There are a couple of possible explanations for this. First, the 9-ANS element model is coarser than the 410 element model — there are fewer elements and less interpolating power — even though it involves more degrees of freedom (due to the unavoidable extraneous nodes in the stiffener cross-sections). Thus, refinement of the 9-ANS model may reduce the discrepancy with experiment. Second, due to differences in the accuracy of the eigensolutions for the 410 and 9-ANS models, the imperfections are not *exactly* the same; small differences in the imperfections can produce large differences in response for such imperfection-sensitive structures. Finally, note that the imperfections used were not experimentally measured, but rather "pulled out of the air", and may just happen to produce more accurate results for the 410 model than for the 9-ANS model.

These and other issues are now being studied, in addition to another, related focal problem, which features a *curved* panel, and for which experimentally measured imperfections are provided.

PITFALLS

- **Mixed Precision**
- **DOF Renumbering Req's for Sparse Solvers ($10^7 \rightarrow 10^6$ storage)**
- **Specified Displacements and Arc-Length Methods**
- **Data Structures**
  - **— Overwriting essential for Nonlinear Analysis**
  - **— Should make more nominal — eventually**

PITFALLS

Some important lessons were learned during the course of analyzing the preceding problems. The following is a partial list:

i) One should not risk using single precision for any CSM calculations on 32-bit word computers (such as VAXes and SUNs):

ii) Degree-of-freedom renumbering should always be performed in conjunction with sparse-matrix equation solvers, such as those currently used in the Testbed. For example, even though the initial node numbers for the I-stiffened panel models would have been nearly optimal for band and profile solvers, renumbering for *fill minimization* (via the Nested Dissection method) resulted in an order-of-magnitude decrease in required storage space for the factored matrix, and a similar reduction for factorization and solution times;

iii) Specified displacement loading must be added with care to arclength-type nonlinear static solution algorithms (more about this in the next chart):

iv) The current Testbed global data structures need revision. First, most datasets should use smaller granularity named (nominal) records so that the GAL write-in-place option can always be used; this is especially important for nonlinear analysis since large quantities of data are being written and re-written during the iteration process. Second, more *relational* data structurs should be adopted to facilitate pre- and post-processing, either interactively or via interface programs. In spite of these recommendations, we think that the transition to improved data structures should occur slowly, to permit the CSM user/developer community to adapt.

SPECIFIED DISPLACEMENTS AND

Arc-Length Methods

Lockheed Palo Alto *CSM Testbed Development*

[Crisfield]

$$\|\Delta \mathbf{d}\|^2 + s\,\Delta\lambda^2 \|\mathbf{f}_0\|^2 = \Delta L^2$$

$$\Delta \mathbf{d} = \begin{Bmatrix} \Delta \mathbf{d}^u \\ \Delta \mathbf{d}^s \end{Bmatrix} \Longrightarrow$$

$$\boxed{\|\Delta \mathbf{d}^u\|^2 + \Delta\lambda^2 \|\mathbf{d}_0^s\|^2 = \Delta L^2}$$

ORIGINAL PAGE IS
OF POOR QUALITY

SPECIFIED DISPLACEMENTS AND
Arc-Length Methods

Lockheed Palo Alto *CSM Testbed Development*

This chart highlights one of the pitfalls mentioned on the previous chart: introducing specified displacement loading into arclength-type solution algorithms, such as Crisfield's algorithm [12]. While Crisfield's arclength constraint equation omits the specified force ($\mathbf{f}_0$) loading term — to avoid scaling (s) problems — the analogous term for specified displacement loading must be included. This term can be turned on automatically by including the specified components of incremental displacement, $\Delta\mathbf{d}^s$, when taking the displacement norm, $\|\Delta\mathbf{d}\|$. Note that these components should be left out of other incremental displacement vectors (for example, when computing error norms), since the specified displacement components are, by definition, not considered as unknowns.

We have found — by experience — that if the $\Delta\lambda^2\|\mathbf{d}_0^s\|^2$ term is not included in the arclength (ΔL) constraint equation (last equation in the present chart), it is often not possible to drive the solution over limit points. This of course defeats the main purpose for using an arclength method in the first place. Note that such subtle points are not easily found in the literature.

CONCLUSIONS

- **CSM Testbed Increasing our Productivity:**
  - **— Teamwork**
  - **— Interaction**
  - **— Useful Results**
- **Computational Efficiency not a Serious Drawback**
- **Generic Element Processor Proving Successful at LPARL**
- **Work needed in 3 other areas ...**

CONCLUSIONS

As a result of our increasing use and development of the CSM Testbed, our own research laboratory is experiencing an increase in productivity as well. For the first time, we are able to work together via a common software interface, to interact on diverse tasks (when appropriate), and to produce results that are of visible use to others.

Furthermore, one of our fears regarding the Testbed has been alleviated: computational efficiency. Comparisions with the STAGS code yield run times within ±50% (depending on function), for problem sizes on the order of 5000 degrees-of-freedom. More statistics are now being gathered for benchmark documentation.

Regarding the generic element Processor, it does seem to expedite the introduction of new elements into the Testbed; for example the STAGS 410 element was introduced in about a week (once the necessary strings were cut from the STAGS code). Even less time for new-element entry is anticipated now that some of the bugs have been removed, and tutorial documentation (in preparation) should facilitate the process at remote sites.

Nevertheless, more work is needed both on the generic element Processor and in three other areas, as discussed in the following two charts.

PLANS

Generic Element Processor

Lockheed Palo Alto — *CSM Testbed Development*

- Documentation
- Advanced Arc-Length Algorithm [Riks] – NAVY
- More on Beam and Solid Elements

- Test Dynamics
- Constraint (e.g., Contact) Elements
- Boundary Elements
- Non-Structural Elements: ET* EF*
- ERROR CONTROL
- Vectorization
- PARALLELIZATION: ES1 ES1 ES2 ...

PLANS

Generic Element Processor

Lockheed Palo Alto

CSM Testbed Development

Development of the Generic Element Processor should be an ongoing process, with extensions added to accommodate the items mentioned in the present chart. Of particular importance, is the introduction of error control features, which will commence with the implementation of a family of p-refinement elements supplied by Barna Szabo. Such extensions, however, will not be isolated within the element Processor; for example, new high-level *Procedures* will have to be developed for adaptive refinement; and to exploit the p-refinement technique, new matrix utilities will be required as well (see next chart).

Another major thrust will be in the area of parallel processing, where independent element Processors will be mapped to hardware processors, probably according to a substructure partitioning scheme such as that proposed by Nour-Omid and Ortiz [16] in their "cut and paste" solution algorithm. Note that internal *vectorization* of element Processors is also on the agenda, having formulated a tentative plan for providing element kernels with "blocks" of element data instead of one element at a time. This would also provide a mechanism for automatic vectorization of element developer's code, should the need arise.

Finally, the present generic (structural) finite element Processor "shell" concept could easily be extended to non-structural elements.* For example, Generic Fluid Element Processors (EF*) and Generic Thermal Element Processors (ET*) would be useful in constructing procedures for structure-media interaction problems. Similarly, boundary elements would warrant *Generic Boundary Element Processors* for structural and non-structural applications (e.g., BES*, BEF* and BET*). The rationale for such subdivisions is the difference in the high-level command language and data structures natural for each domain (e.g., EF* Processors would probably have `FORM VISCOSITY` and `FORM CONVECTION` commands, respectively; and while BES* Processors would have the same commands as ES* Processors, they would probably output full, non-symmetric assembled matrices rather than a sequence of symmetric element matrices).

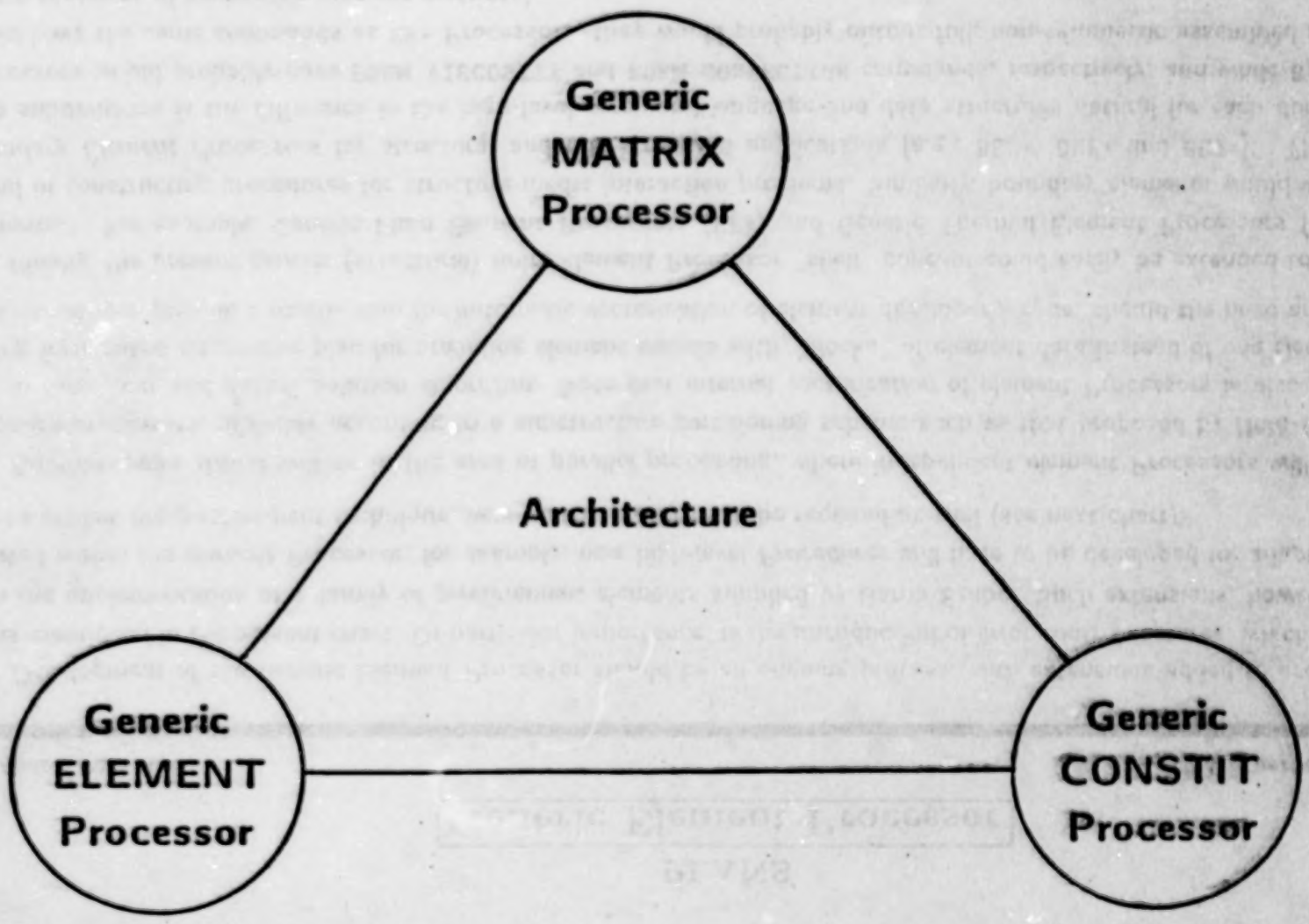
---

* The Generic Element Processor design removes the burden of *how* element arrays are to be computed from the algorithm developer, who is primarily interested in *what* should be formed. This (message-passing) feature and the shell/kernel (inheritance) aspect of the design have much in common with so-called object-oriented programming languages, which are intended to simplify and extend the life-cycle of software development efforts [17].

PLANS

CSM Testbed

Lockheed Palo Alto — *CSM Testbed Development*



PLANS
CSM Testbed

Having gained some experience with the Generic Element Processor, we believe that it suggests an overall strategy for long-term development of the Testbed. Specifically, we believe that *three* classes of generic Processors should be available: Element, Constitutive and Matrix. This triad can be viewed as a higher-level extension of the truly generic Testbed architecture (NICE) — but with more leverage towards CSM-specific applications. For it is one of the attributes of CSM (say, versus CFD) that it can be neatly partitioned into these three areas. Element developers, matrix algebraists (e.g., solution algorithm developers) and material scientists (e.g., composite mechanicians) can, for the most part, work independently of one another. Yet, all three types of researchers can voluntarily work together — on the same problem — if and when they're ready, via the integration provided by the underlying architecture.

(Imagine a numerical analyst testing a new eigensolution algorithm on a 10×10 manually entered test matrix; a material scientist simultaneously checking out a new high-temperature visco-plastic constitutive model for metal-matrix composites, at a single spatial integration test point; and an element developer independently deriving a new family of hierarchically refinable shell elements validating his/her elements via the MacNeal/Harder test cases — plus others mentioned herein. Next imagine these three researchers stepping aside and letting an engineering analyst apply this triad of combined new capabilities to a complex model of the NASP airframe — without software modifications. This is the envisioned role of the Testbed.)

We are presently working on a plan for the Generic *Matrix* Processor, which would allow an arbitrary number of alternate, or complementary, matrix utilities (and data structures) to be implemented within the Testbed. Importantly, the generic Processor "shell" for these developer-supplied utilities (i.e., "kernels") would feature a common, but extendible, data-structure driven matrix algebraic language, which would enable the user to solve equations via a variety of techniques. This should greatly facilitate the current CSM research in parallel processing — *within* the Testbed.

REFERENCES

[1] Felippa, C.A. and Stanley, G.M., "NICE: A Utility Architecture for Computational Mechanics", in proceedings of *Finite Element Methods for Nonlinear Problems*, eds. Bergan, Bathe and Wunderlich, Europe-US Symposium, Trondheim, Norway, 1985.

[2] Whetstone, W.D., *SPAR Structural Analysis System Reference Manual*, Vol. 1: NASA CR 158970-1, Dec. 1978.

[3] Felippa, C.A., "The Computational Structural Mechanics Testbed Architecture: Volume IV – The Global-Database Manager GAL-DBM", NASA Contractor Report 178387, 1987.

[4] Felippa, C.A., "The Computational Structural Mechanics Testbed Architecture: Volume I – The Language: Volume II – The Directives: Volume III – The Interface", NASA Contractor Report 178384–6, 1987.

[5] Belytchko, T. and Hsieh, B.J., "Non-Linear Transient Finite Element Analysis with Convected Coordinates", *IJNME*, 7, 255–271, 1973.

[6] Wempner, G., "Finite Elements, Finite Rotations and Small Strains of Flexible Shells", *International Journal for Solids and Structures*, 5, 117–153, 1969.

[7] Rankin, C.C., "Consistent Linearization of the Element-Independent Corotational Formulation for the Structural Analysis of General Shells", NASA Contractor Report 278428, 1988.

[8] Park, K.C. and Stanley, G.M., "A Curved C^0 Shell Element Based on Assumed Natural-Coordinate Strains", *Journal of Applied Mechanics*, 53, 278–290, 1986.

[9] Stanley, G.M., "Continuum-Based Resultant Shell Elements", in *Finite Element Methods for Plate and Shell Structures, Vol. 1: Formulations and Algorithms*, eds. T.J.R. Hughes and E. Hinton, pp. 1–45, 1986.

[10] Kang. D.S. and Pian. T.H.H.. "A Versatile and Low Order Hybrid Stress Element for General Shell Geometry". *AIAA.* 87-0840. 633–641. 1987.

[11] Almroth. B.O.. Brogan. F.A. and Stanley. G.M.. "Structural Analysis of General Shells. Vol. II: User instructions for the STAGS(C-1) computer code". Lockheed Report LMSC-D33873. 1979.

[12] Crisfield. M.A.. "A Fast Incremental/Iterative Solution Procedure that Handles Snap-Through". *Computers and Structures.* 13. 55–62. 1983.

[13] MacNeal. R.H. and Harder. R.L.. "A Proposed Standard Set of Problems to Test Finite Element Accuracy". *Journal of Finite Elements in Analysis and Design.* 1. 3–20. 1985.

[14] Thurston. G.A.. Brogan. F.A. and Stehlin. P.. "Postbuckling Analysis Using a General Purpose Code". *AIAA Journal.* 24/6. 1013–1020. 1986.

[15] Starnes. J.H. Jr.. Knight. N.F. Jr. and Rouse. M.. "Postbuckling Behavior of Selected Flat Stiffened Graphite-Epoxy Panels Loaded in Compression". *AIAA Journal.* 23/8. 1236–1246. 1985.

[16] Ortiz. M. and Nour-Omid. B.. "Unconditionally Stable Concurrent Procedures for Transient Finite Element Analysis". *Computer Methods in Applied Mechanics and Engineering.* 58. 151–174. 1986.

[17] Cox. B.J.. *Object Oriented Programming – An Evolutionary Approach.* Addison-Wesley. 1986.

N89
29796
UNCLAS

N89-29796

57-39
308
211964
AV 025068

ASSESSMENT OF SPAR ELEMENTS AND FORMULATION OF SOME BASIC 2-D AND 3-D ELEMENTS FOR USE WITH TESTBED GENERIC ELEMENT PROCESSOR

MOHAMMAD A. AMINPOUR+

INTRODUCTION

Computational Structural Mechanics (CSM)

The initial CSM Testbed was based on Level 13 of the SPAR finite element computer program. Until recently, the element library of the Testbed has been limited to those elements in Level 13 of SPAR. The development of a generic element processor has enabled element researchers to develop, implement and assess element formulations with relative ease. An assessment of new elements as well as the existing SPAR Level 13 elements has revealed some definite shortcomings with the SPAR Level 13 2-D and 3-D elements. The SPAR S81 solid element does not pass the patch test problem proposed by MacNeal-Harder[1]. These deficiencies are identified here. ~~below.~~ The 2-D elements, however, seem to perform well taking into account the limitations imposed by the theory used to formulate them, (i.e., thin plates only). Common deficiencies of the 2-D and 3-D elements in SPAR have to do with their adaptability to the nonlinear analysis utilities developed by Lockheed Palo Alto Research Lab. Also, the "EFIL" format of the SPAR element data does not conform to the standard format of the Testbed. Accordingly, even if the deficiencies of the 3-D elements concerning their performance are corrected, their use will still be limited in the Testbed environment. Therefore, a new set of some basic 2-D and 3-D elements have been formulated to overcome the performance deficiencies as well as the deficiencies concerning their adaptability to nonlinear analysis utilities in the Testbed and the conformity of their EFIL to the standard format of the Testbed. These elements are formulated and implemented into the Testbed as two element processors (processors ES3 and ES4). Finally, the performance and accuracy of these elements are studied.

+Research Scientist, Analytical Services and Materials, Inc., Hampton, VA.

PRECEDING PAGE BLANK NOT FILMED

PAGE 652 INTENTIONALLY BLANK

OBJECTIVES

- ASSESS SPAR ELEMENTS AND IDENTIFY DEFICIENCIES
- FORMULATE NEW 2-D AND 3-D ELEMENTS
  - STANDARD EFIL FORMAT
  - PROVIDE EASY USE OF NONLINEAR ANALYSIS UTILITIES
- UTILIZE GENERIC ELEMENT PROCESSOR
- ASSESS NEW ELEMENTS

OBJECTIVES

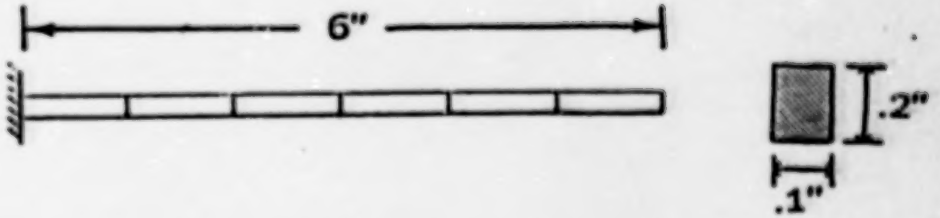
The initial objectives of this study were to assess the accuracy and performance of the 2-D and 3-D elements in SPAR Level 13 for linear stress analysis and to identify any deficiencies, and to modify the elements, if necessary. However, after studying the elements, it was decided to develop a new set of 2-D and 3-D elements in the Testbed using a different element formulation than the element formulation in SPAR. The elements in SPAR are based on the Pian hybrid stress formulation[2]. These new Testbed elements are developed based on the Reissner mixed formulation for reasons that will be discussed later. The new elements have a standard EFIL format, i.e., the EFIL format of the generic element processors developed by Lockheed Palo Alto. The new elements provide flexibility for finite element research and also provide flexibility for use of nonlinear analysis utilities developed by Lockheed Palo Alto. Finally, the performance and accuracy of new elements are assessed.

SPAR 1-D AND 2-D ELEMENTS RESULTS

MACNEAL-HARDER TEST CASE RESULTS

STRAIGHT CANTILEVERED BEAM

RECTANGULAR-ELEMENTS



Normalized Tip Displacement in Direction of Load †

Tip Loading Direction	SPAR element E21 (1-D)	SPAR element E43 (2-D)	QUAD 4 MSC/ NASTRAN
Extension	1.000	.996	.995
Inplane Shear	.998	.993	.904
Out of Plane Shear	.998	.986	.986
Twist	.938	.664	.941

† Exact solutions are given in Ref. 1.

SPAR 1-D AND 2-D ELEMENTS RESULTS

The test problem selected to verify the performance of the 1-D and 2-D elements in SPAR Level 13 is shown on the opposite page. This problem has been proposed by MacNeal-Harder[1] as a test case. It is a straight cantilever beam discretized into six finite elements. The beam is 6.0 in. long and 0.2 in. deep. The width of the beam is 0.1 in. These dimensions define a rather long beam and the discretization is quite coarse, making it difficult to achieve the correct solution. The beam is subjected to different types of tip loading, and the normalized displacements at the tip in the direction of the load are tabulated in the table shown on the opposite page. The finite element solutions are normalized by the exact solutions as presented in Ref. 1. The SPAR E21 beam element achieves good results with a maximum error of 6.2% for the case of twist (torsion) loading. In the case where the SPAR E43 flat shell element (class C^1 continuity) is used, the results are all satisfactory except for the case of twist. The reason for this discrepancy is that the SPAR 2-D elements are formulated according to Kirchhoff's thin plate theory in which the effect of transverse shear deformations are ignored. It is observed that this is a thick plate, the thickness being half the height of the beam. When the beam is loaded in bending the effect of shear deformation is very small (negligible) because of the long length of the beam as compared with the other dimensions. However, when the beam is loaded in torsion, the effect of the transverse shear deformation becomes important for a thick plate, like the one considered here. Therefore, the SPAR E43 element performed as expected, i.e., to perform well under bending type of loading and not nearly as well under torsion type of loading for this problem. Results using another 4-node quadrilateral element, namely the MSC/NASTRAN QUAD4 element are also presented in the table for comparison. Using the QUAD4 element, all results have a maximum error of less than 10%. These results are from Ref. 1, which also includes results using other MSC/NASTRAN elements.

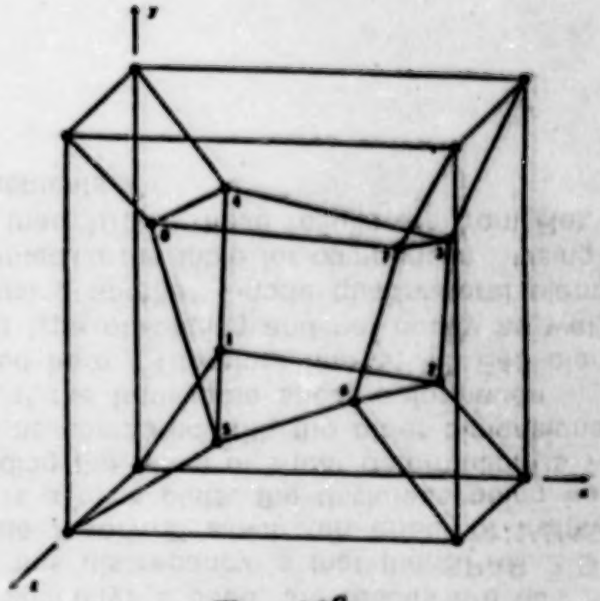
TEST PROBLEM FOR SPAR S81 ELEMENT

APPLIED DISPLACEMENTS

$$u = \frac{1}{2}(2x + y + z) \times 10^{-3}$$

$$v = \frac{1}{2}(x + 2y + z) \times 10^{-3}$$

$$w = \frac{1}{2}(x + y + 2z) \times 10^{-3}$$



$E = 10^6$

$\nu = .25$

RESULTING STRAINS AND STRESSES

$$\varepsilon_{xx} = \varepsilon_{yy} = \varepsilon_{zz} = 10^{-3}$$

$$\gamma_{xy} = \gamma_{yz} = \gamma_{xz} = 10^{-3}$$

$$\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = 2000$$

$$\sigma_{xy} = \sigma_{yz} = \sigma_{xz} = 400$$

TEST PROBLEM FOR SPAR S81 ELEMENT

The test problem selected to verify the performance of the 3-D elements in SPAR Level 13 is shown on the opposite page. This problem has been proposed by MacNeal-Harder[1] as a patch test for solid elements. The problem is composed of seven hexahedron elements forming a unit cube. All the faces of the elements, except for the faces forming the external boundary, are warped (i.e., the four points of a face do not lie in one plane). The locations of the internal points are given in Ref. 1. The suggested displacement boundary conditions on the external boundary are stated on the opposite page. This set of displacement boundary conditions results in a constant strain loading condition with the resulting stresses as shown. The SPAR S81 solid element was used to solve this problem. Unfortunately the computer run failed to complete because the strain energy matrix turned out to be singular, at least, for some of the elements. When the faces of the middle hexahedron were made flat so that no warping existed on the internal element faces, the fate of the computer run did not change. Therefore, investigations were conducted regarding the formulation and implementation of the SPAR S81 solid element.

ASSUMED STRESS HYBRID FORMULATION

$$\Pi = \frac{1}{2} \int_V \sigma^T D^{-1} \sigma \, dv - \int_S (n^T \sigma)^T u \, ds + \int_{S_T} u^T t_o \, ds$$

ASSUMPTIONS: Equilibrium is satisfied

strains derived from stresses

continuous displacements

$$T = \int_S (n^T \sigma)^T u \, ds = \int_V \sigma^T L[u] \, dv + \underbrace{\int_V (L^T[\sigma])^T u \, dv}_{= 0}$$

ASSUMED STRESS HYBRID FORMULATION

The SPAR finite element formulation is based on the principal of minimum complementary energy. The formulation is implemented in accordance with the Pian assumed stress hybrid formulation[2], the functional for which is given on the opposite page[3,4]. For this functional to be valid, the assumptions stated on the opposite page must hold true when choosing the form of the stresses, strains, and displacements[4]. The assumed stresses are treated as the field variables and the displacements are treated as the interface (hybrid) variables on the boundary of the elements. Furthermore, the assumed displacements must satisfy the displacement boundary conditions[4]. When these assumptions hold true, the minimization of the functional recovers, in an integral sense, the strain-displacement relationship, forces the tractions to be continuous across the boundaries of adjacent elements, and satisfies the traction boundary conditions[4]. The second integral in the functional, denoted by T, at the bottom of the opposite page is a surface integral representing the work done on the boundaries of the elements. This surface integral can be written in terms of two volume integrals using Gauss's Theorem which is often referred to as integration by parts. As denoted on the opposite page, the second volume integral vanishes for the Pian formulation because the stresses are chosen so that equilibrium is satisfied, i.e., $L^t[\sigma]=0$. Therefore, only the first volume integral needs to be evaluated for each element. This approach, in general, is more efficient than evaluating six surface integrals (one integral per face) for each element. However, the investigations revealed that SPAR Level 13 evaluates the surface integral. Furthermore, it was found that SPAR Level 13 evaluates the unit normal to each surface only once, i.e., SPAR assumes that the surfaces of the elements are flat planes and no warping exists. This feature may explain in part the reason for the failure of the SPAR S81 element for the problem with warped faces. Further studies revealed that the assumed displacements on the boundaries (faces) of the adjacent elements were not continuous, therefore, leaving "gaps" between the adjacent elements and violating the third assumption stated on the opposite page. This explains the failure of the SPAR S81 element for the test problem with flat faces. However, the assumed displacement functions for the S81 element are continuous across the element boundaries (faces) only when the elements are rectangular parallelopiped, and the more the deviation from a rectangular parallelopiped, the larger the error. Therefore, for slight distortions, say 10 degrees, the SPAR S81 element will work, but for MacNeal-Harder problem with distortions of the order 45 degrees the SPAR S81 element fails to work.

DEFICIENCIES NOTED FOR SPAR ELEMENTS

- INTERELEMENT DISPLACEMENT CONTINUITY REQUIREMENT VIOLATED
- SENSITIVE TO OUT-OF-PLANE WARPING
- DIFFICULT TO COUPLE WITH GENERIC ELEMENT PROCESSOR
- DIFFICULT TO COUPLE WITH NONLINEAR ANALYSIS UTILITIES
- COMPUTATIONALLY INEFFICIENT
- LACK OF IN-LINE DOCUMENTATION; DIFFICULT TO MAINTAIN OR UPGRADE
- NONSTANDARD EFIL FORMAT

DEFICIENCIES NOTED FOR SPAR ELEMENTS

The first two deficiencies listed on the opposite page are peculiar to the SPAR solid elements S81 and S61 and have already been discussed. The other deficiencies listed are common to both 2-D and 3-D elements in SPAR. These deficiencies are self explanatory and were discussed, to a limited extent, in the introduction.

PROPOSED SOLUTION TO DEFICIENCIES

- DEVELOP NEW 2-D AND 3-D ELEMENTS
  - NEW FORMULATION (REISSNER MIXED METHOD)
  - IMPROVED COMPUTATIONAL EFFICIENCY
  - EASY TO FOLLOW AND MAINTAIN FOR ELEMENT RESEARCH
- COMPATIBILITY WITH GENERIC ELEMENT PROCESSOR
- STANDARD EFIL FORMAT
- PROVIDES FOR NONLINEAR ANALYSIS CAPABILITY

PROPOSED SOLUTION TO DEFICIENCIES

It is desirable to modify the SPAR element architecture to overcome the deficiencies regarding their compatibility with generic element processor, standard EFIL format, and their easy adaptability for use with nonlinear utilities. It is desirable that the solid elements not be limited for use as rectangular parallelopiped elements and be able to perform well for elements with warped faces. With these considerations in mind, some basic 2-D and 3-D elements have been formulated based on the Reissner mixed method functional so that the above objectives are accomplished. Also, the new implementation would be computationally more efficient and easy to follow, maintain, and upgrade for future element research.

PRESENT FORMULATION BASED ON

REISSNER MIXED METHOD

$$\Pi = \frac{1}{2} \int_V \sigma^T D^{-1} \sigma \, dv - \int_V \sigma^T L[u] \, dv + \int_{S_T} u^T t_o \, ds$$

ASSUMPTIONS: Strains derived from stresses

Continuous displacements

PRESENT FORMULATION BASED ON REISSNER MIXED METHOD

As discussed earlier, in order to reach the stated objectives, some basic 2-D and 3-D elements are formulated using the Reissner mixed method. The functional for the Reissner mixed method, along with assumptions to make this functional valid, are given on the opposite page[4]. Two assumed independent field variables are in this formulation. One field is either the stresses or the strains, and the other field is the displacements. The assumed displacements must also satisfy the displacement boundary conditions[4]. The minimization of this functional recovers, in an integral sense, the equations of equilibrium, strain-displacement relations, forces the tractions to be continuous across the boundaries of adjacent elements, and finally satisfies the traction boundary conditions[4]. A quick look at the Pian hybrid stress formulation and the accompanied discussion reveals that had the surface integral, which was denoted by $\bar{T}$, been replaced with its equivalent volume integral and had the displacements been treated as field variables instead of interface variables, the present formulation would have been obtained. When the assumed stresses in the present functional are chosen to satisfy the equilibrium equations, this functional will, as it must, reduce to that of the principal of minimum complementary energy, which is the basis for the hybrid stress formulation. The present formulation based on Reissner mixed method has certain advantages which will be discussed next.

ADVANTAGES OF REISSNER MIXED METHOD

- IF STRESS FUNCTIONS CHOSEN SATISFY EQUILIBRIUM, THEN FORMULATION REDUCES TO PIAN HYBRID-STRESS FORMULATION

- SINCE ASSUMED FIELD DOES NOT HAVE TO SATISFY EQUILIBRIUM, THEN FORMULATION MAY:
  - ASSUME STRESS OR STRAIN FIELDS
  - EXPRESS ASSUMED FIELD IN CARTESIAN OR NATURAL COORDINATES

- THIS FREEDOM OF CHOICE IS USEFUL FOR ELEMENT RESEARCH AND NONLINEAR ANALYSIS

ADVANTAGES OF REISSNER MIXED METHOD

One of the merits of this method is the one already discussed, i.e., if the assumed stresses are chosen so as to satisfy the equilibrium equations then the formulation will, as it must, reduce to the hybrid stress formulation. The other merit of this formulation is that since the assumed stresses now do not have to satisfy the equilibrium equations, one has more freedom in assuming the form of the stress functions. For example, strain functions can be assumed and the stresses can be calculated from the assumed strains. In general, these stresses will not satisfy the equilibrium equations, but in this formulation we have the freedom to do so. This freedom to assume the strains rather than the stresses is useful for material nonlinear analysis, e.g., plasticity. In addition, Pian points out[5] that for deformed elements it would be better if one chooses the stresses (or strains) in Lagrangian natural coordinates rather thar. the rectangular Cartesian coordinates as this will eliminate the directional bias in assuming the form of the stress or strain functions. Therefore, this implementation provides the freedom to choose the stresses (or strains) in either coordinates and to study their relative performance.

NEW ELEMENTS IN THE TESTBED

- SOLID ELEMENTS (PROCESSOR ES3)

8-NODE HEXAHEDRON	20-NODE HEXAHEDRON
6-NODE PENTAHEDRON	15-NODE PENTAHEDRON
4-NODE TETRAHEDRON	10-NODE TETRAHEDRON

- FLAT SHELL ELEMENTS (PROCESSOR ES4)

4-NODE MEMBRANE	3-NODE MEMBRANE
4-NODE PLATE (BENDING)	3-NODE PLATE (BENDING)
4-NODE MEMBRANE & BENDING	3-NODE MEMBRANE & BENDING
4-NODE SHEAR PANEL	3-NODE SHEAR PANEL

- ALL ELEMENTS ARE OF CLASS C^0 CONTINUITY

NEW ELEMENTS IN THE TESTBED

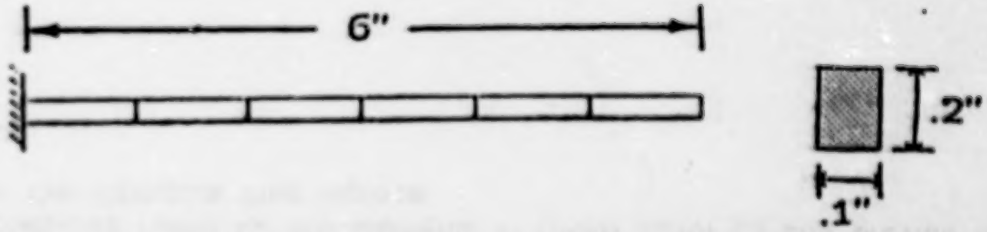
Based on the formulation just described, six solid elements and eight flat shell elements were implemented into the Testbed. The elements are all of class C^0 continuity. The 2-D elements account for the effects of transverse shear deformation, so that problems involving moderately thick plates and laminated composite plates, for which $G_{12}/E_1 < 1/10$, can be solved. This feature is in contrast to the SPAR 2-D elements which are of class C^1 continuity and do not account for the effects of transverse shear deformation, which limits their use to thin plates only. These new elements are implemented in the Testbed as two new processors, processors ES3 and ES4. Processor ES3 contains the six solid elements listed on the opposite page. Processor ES4 contains the eight 2-D flat shell elements listed on the opposite page. These new elements resolve the deficiencies noted for the elements in SPAR Level 13 and provide equivalent or greater capability than the elements they replace.

SHELL ELEMENTS (PROCESSOR ES4) RESULTS

MACNEAL-HARDER TEST CASE RESULTS

STRAIGHT CANTILEVERED BEAM

RECTANGULAR-ELEMENTS



672

NORMALIZED TIP DISPLACEMENT IN DIRECTION OF LOAD

Tip Loading Direction	Assumed Displ.	CARTESIAN COORDINATES		NATURAL COORDINATES		QUAD4 MSC/ NASTRAN
		Assumed Stress	Assumed Strain	Assumed Stress	Assumed Strain	
Extension	.995	.996	.995	.996	.995	.995
Inplane Shear	.0933	.993	.904	.993	.904	.904
Out of Plane Shear	.0302	.980	.980	.980	.980	.986
Twist	.932	.941	.940	1.089	.940	.941
Inplane Moment	.0933	1.000	.910	1.000	.910	—

SHELL ELEMENTS (PROCESSOR ES4) RESULTS

One problem selected to assess the new flat shell elements implemented in Testbed processor ES4 is the MacNeal-Harder[1] cantilever test problem discussed earlier. The solution to this problem is obtained using all the available options in the processor ES4 and the results are tabulated on the opposite page. The results from a conventional isoparametric assumed displacement finite element formulation (class C^0 continuity, also implemented in processor ES4 as an option) performs well for extension, relatively well for torsion, and very poorly for the other types of loading. The performance of the displacement based element is related to the assumed field. Deformation due to extension is represented by the assumed bilinear displacement shape functions exactly, and therefore even one element (instead of six) would provide an accurate result. For the case of deformation due to torsion, the assumed bilinear displacement shape functions represent a fairly good approximation to the deformation, and therefore even with this coarse mesh the result is satisfactory with 6.2% error. On the other hand, deformation due to out-of-plane and in-plane shear varies cubically along the length of the beam and the deformation due to in-plane moment varies quadratically along the length of the beam. Therefore, the assumed bilinear displacement shape functions cannot represent the deformation unless the mesh is refined considerably. The results from MSC/NASTRAN QUAD4 element are reproduced from Ref. 1 for comparison with the current ES4 results using 4-node quadrilateral elements. The QUAD4 element is also an isoparametric element but in addition it employs selective reduced order integration. To account for transverse shear deformations, the QUAD4 element uses string-net approximation and augmented shear flexibility[6]. The results from the implementation of the Reissner mixed method in Testbed processor ES4 (class C^0 continuity) are obtained by assuming the stresses in Cartesian coordinates, the strains in Cartesian coordinates, the stresses in natural coordinates and the strains in natural coordinates. The results are all within a maximum error of less than 10%. Satisfactory performance of these elements with such a coarse mesh is only achieved with rectangular elements. If the elements are distorted, the mesh has to be refined considerably to achieve comparable results (except for the case of extension). For a distorted mesh, the SPAR E43 flat shell elements perform poorly for membrane loading case (except for extension). The E43 element with a distorted mesh has approximately the same performance as the undistorted mesh for bending and torsion loading cases. These results are obtained since the formulation of the SPAR 2-D element for the in-plane deformation is of class C^0 continuity while for the out-of-plane deformation is of class C^1 continuity. However, as mentioned before, class C^1 has the limitation of having satisfactory performance only for thin plates.

SOLID ELEMENTS (PROCESSOR ES3) RESULTS

MACNEAL - HARDER PATCH TEST FOR SOLIDS

STRESS DISTRIBUTION

Element Type	CARTESIAN COORDINATES		NATURAL COORDINATES	
	Assumed Stress*	Assumed Strain	Assumed Stress	Assumed Strain
8-node	exact	exact	exact	exact
20-node	exact	exact	exact	exact

* assumed stresses satisfy equilibrium within each element.

SOLID ELEMENTS (PROCESSOR ES3) RESULTS

The first test problem selected to assess the solid elements developed in the Testbed processor ES3 is the MacNeal-Harder[1] proposed patch test problem for solid elements discussed earlier. Both the 8-node and the 20-node hexahedron elements were used to obtain the solution to this problem. Also all the options available in the processor were used to obtain the solution. The stress functions chosen for these elements are the ones given in Ref. 4 except for the correction of typographical errors. The assumed stresses expressed in Cartesian coordinates satisfy the equilibrium equations, while the assumed strains in Cartesian coordinates, the assumed stresses and the assumed strains in Lagrangian natural coordinates do not satisfy the equilibrium equations. It is worth mentioning again that all the faces (except the ones representing the external boundary) of the elements are warped (i.e., the four corner points of a face do not lie in one plane). In addition, the midside nodes of the 20-node elements were chosen so as to make all the faces (except the ones representing the external boundary) of the elements to be curved surfaces. Therefore, the 8-node and the 20-node elements are being tested to their full capacity. This test would make the hidden errors more likely to show up. For all cases however, the exact theoretical stress distribution was recovered.

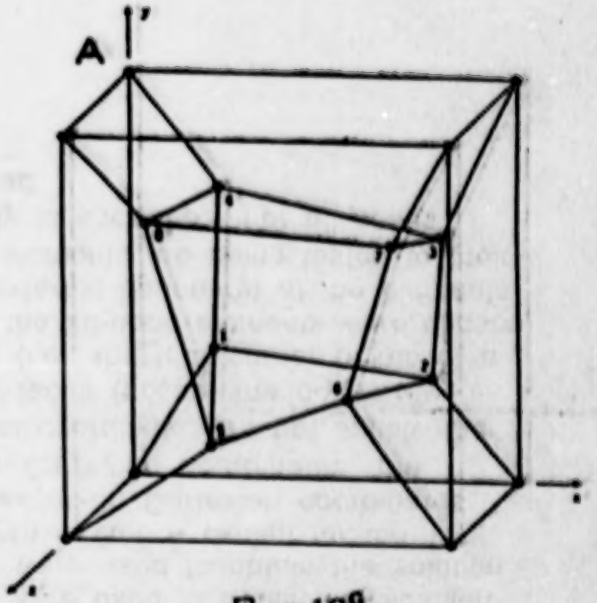
ANOTHER TEST PROBLEM FOR SOLID ELEMENTS

APPLIED STRESSES

$$\sigma_{xx} = +1000x$$

$$\sigma_{xy} = -1000y$$

$$\sigma_{yy} = \sigma_{zz} = \sigma_{yz} = \sigma_{zz} = 0$$



$E = 10^6$

$\nu = .25$

RESULTING DISPLACEMENTS

$$u = \frac{1}{2}\left(x^2 - 2.25y^2 + .25z^2\right) \times 10^{-3}$$

$$v = -.25xy \times 10^{-3}$$

$$w = -.25xz \times 10^{-3}$$

ANOTHER TEST PROBLEM FOR SOLID ELEMENTS

As a second test to assess the performance of the solid elements in processor ES3 of the Testbed, the same mesh was chosen with a different set of boundary conditions. The tractions applied to the boundaries of the solid and the resulting theoretical displacements are stated on the opposite page. The maximum displacement will occur at point A (x=z=0 y=1) in the X-direction. This problem is different from the previous one in that it is not a constant strain problem. In fact, this problem presents a very severe boundary condition for the 8-node element. For this element, the assumed σ_{xx} is a function of a local element coordinate system y and z only[7], while the assumed σ_{xy} is a function of z only[7], of the local coordinate system. In other words, the assumed stress functions for the 8-node element do not represent the traction boundary conditions applied to this problem. Therefore, the 8-node element is not expected to have a superb performance for this problem, unless the mesh is refined. On the other hand, the 20-node element is expected to have a satisfactory performance because the assumed σ_{xx} and σ_{xy} functions[7] do represent the traction boundary conditions applied to this problem.

SOLID ELEMENTS (PROCESSOR ES3) RESULTS

NORMALIZED DISPLACEMENT AT POINT A IN THE x DIRECTION

Element Type	CARTESIAN COORDINATES		NATURAL COORDINATES	
	Assumed Stress*	Assumed Strain	Assumed Stress	Assumed Strain
8-node	.746	.760	.749	.763
20-node	1.005	1.027	1.008	1.066

* assumed stresses satisfy equilibrium within each element.

SOLID ELEMENTS (PROCESSOR ES3) RESULTS

From the resulting displacement functions on the previous page, it is deduced that maximum displacement will occur at point A in the X-direction and has a value of -1.125×10^{-3}. From the table on the opposite page, the 8-node element performs about the same (in error by about 25%) regardless of the assumed field and the coordinate system. However, following the discussion on the previous page, this is to be expected. On the other hand, the 20-node element has a much better performance as expected. A maximum error of 6.6% is observed for the case where the strain field is assumed and expressed in the natural coordinates, while the best performance (.5% error) is attributed to the case where the stress field is assumed and expressed in the Cartesian coordinates to satisfy the equilibrium equations.

SUMMARY

- NEW SOLID AND SHELL ELEMENT FAMILIES
- ELEMENT FORMULATION BASED ON REISSNER MIXED METHOD
- NEW ELEMENTS IMPLEMENTED AS NEW PROCESSORS ES3 (SOLID ELEMENTS) AND ES4 (SHELL ELEMENTS)
- STANDARD EFIL FORMAT FOR ALL ELEMENTS
- NEW ELEMENTS PROVIDE FLEXIBILITY FOR FINITE ELEMENT RESEARCH

SUMMARY

A family of solid elements and a family of flat shell elements have been developed and implemented in the Testbed. They are all of class C^0 continuity. The solid elements are implemented in the Testbed as processor ES3 and the flat shell elements are implemented as processor ES4. The formulation of these elements is based on the Reissner mixed method and the user has the option of choosing either assumed stress or assumed strain field each of which can be expressed in the Cartesian or the Lagrangian coordinates. These elements are implemented in conjunction with the Testbed generic element processor developed by Lockheed Palo Alto and have a standard EFIL format. These elements provide flexibility for the use of the nonlinear analysis utilities developed by Lockheed Palo Alto. These elements also provide flexibility for finite element research in that different functions can be assumed for the field variables (stress or strain) and their performance be studied. This capability is possible, because in this formulation, the assumed field (stress or strain) is not required to satisfy the equilibrium equations.

REFERENCES

1. MacNeal, R. H.; and Harder, R. L.: A Proposed Standard Set of Problems to Test Finite Element Accuracy. Finite Element Analysis and Design, Vol.1, No.1, April 1983, pp. 3-20.

2. Pian, T. H. H.: Derivation of Element Stiffness Matrices by Assumed Stress Distributions. AIAA J., Vol. 2, pp. 1333-36, 1964.

3. Whetstone, W. D.; Yen, C. L.; and Jones, C. E.: SPAR Structural Analysis System Reference Manual. System Level 13A, Vol. 2, Theory, December 1978.

4. Zienkiewiez, O. C.: The Finite Element Method. Third Edition, McGraw-Hill Book Company (UK) Limited, 1977.

5. Pian, T. H. H. : Evolution of Assumed Stress Hybrid Finite Element. Fourth World Congress and Exhibition on Finite Element Methods, The Congress Centre, Interlaken, Switzerland, pp.17-21, September 1984.

6. MacNeal, R. H.: A Simple Quadrilateral Shell Element. Computers and Structures, Vol. 8, pp. 175-183, Pergamon Press 1978.

7. Pian, T. H. H.; and Chen, D.: On The Suppression of Zero Energy Deformation Modes. International Journal for Numerical Methods in Engineering, Vol. 19, 1983, pp. 1741-52.

N89
29797
UNCLAS

N89-29797

S8-39
508.

211965

V16 16109

Development and Verification of Local/Global Analysis Techniques for Laminated Composites[1]

O. Hayden Griffin, Jr.
Department of Engineering Science & Mechanics
Virginia Polytechnic Institute & State University
Blacksburg, Virginia 24061

Introduction.

Analysis and design methods for laminated composite materials have been the subject of considerable research over the past 20 years, and are currently well developed. In performing the detailed three-dimensional analyses which are often required in proximity to discontinuities, however, analysts often encounter difficulties due to large models. Even with the current availability of powerful computers, models which are too large to run, either from a resource or time standpoint, are often required.

There are several approaches which can permit such analyses, including substructuring, use of superelements or transition elements, and the global/local approach. This effort is based on the so-called "zoom" technique to global/local analysis, where a global analysis is run, with the results of that analysis applied to a smaller region as boundary conditions, in a many iterations as is required to attain an analysis of the desired region.

Before beginning the global/local analyses, it was necessary to evaluate the accuracy of the three-dimensional elements currently implemented in the CSM Testbed. It was also desired to install, using the Experimental Element Capability, a number of displacement formulation elements which have well known behavior when used for analysis of laminated composites.

[1] This work is supported by NASA Grant Number NAG-1-672.

OBJECTIVES

- To implement VPI 3-D element family
- To define modeling requirements of laminated composites
- To develop Global/Local analysis procedure

OBJECTIVES

This project has three primary objectives:

1. To implement the family of 3-D elements which have been in use for analysis of composite materials at Virginia Tech for the past 15 years (e.g. see References 1-3)

2. To define the special modeling requirements which exist in laminated composites. This research will provide a consistent set of techniques for 3-D analysis of laminated composites.

3. To develop global/local analysis procedures for laminated composites. The development of procedures for both global and local analyses as well as rules for determining proper location of the global/local interface in laminates are included in this research.

This effort will utilize the CSM Testbed exclusively, except where verification of new capabilities must be checked against previously verified software. The Testbed has been in use at Virginia Tech for this project for approximately one year. All numerical results to date have been generated on a DEC VAX 11/780 computer.

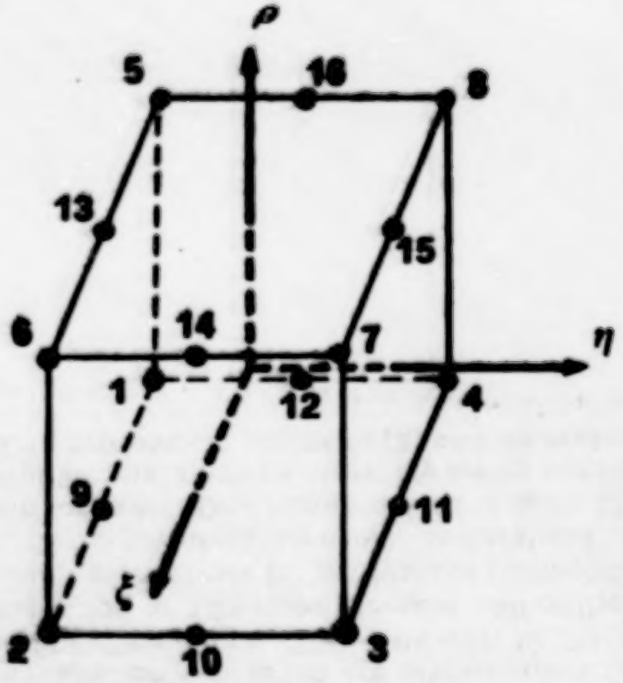
3-D ELEMENTS

- Implemented using Experimental Element Capability of CSM Testbed
- Displacement Formulation
- 16-, 20-, and 32-node elements implemented
- 24-node element implementation in progress
- Elements well proven by use in NASA-VA TECH Composites Program

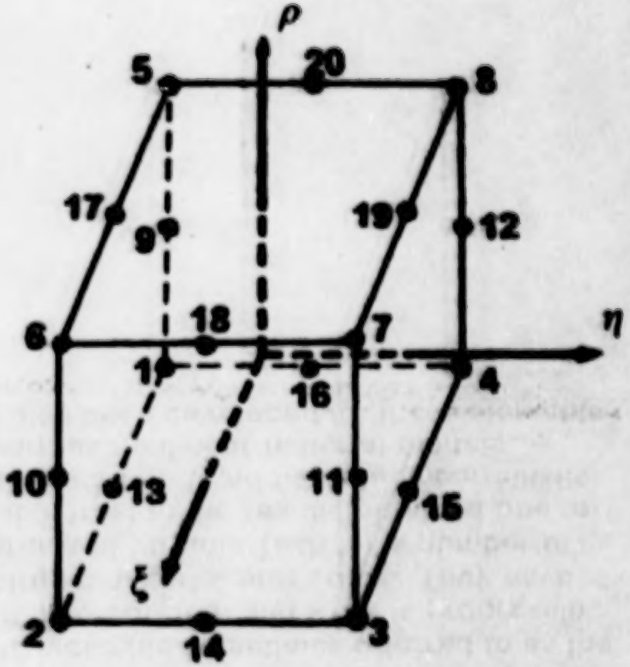
3-D ELEMENTS

Considerable progress has been made to date in enhancing the 3-D analysis capabilities of the CSM Testbed. The approach has been to implement new elements using the Experimental Element Capability of the Testbed. The new elements installed are displacement formulation isoparametric elements sometimes referred to as the "Serendipity" elements. Elements with 16, 20, and 32 nodes have been incorporated, and work is progressing on installation of a 24-node element. All of these elements have orthotropic material properties. They have been used extensively for analysis of composites in the research programs at Virginia Tech for a number of years. Their performance is well understood, and there is a considerable trust in the results obtained due to comparison with other numerical techniques [2] as well as experimental data [4]. In addition to linear elastic constitutive relationships currently being considered for this project, various nonlinear material models, including orthotropic plasticity [2] and an endochronic model [5], have also been developed for these elements.

ELEMENT GEOMETRY



16-Node Element



20-Node Element

ELEMENT GEOMETRY

Two of the element geometries are shown. The 16-node element has eight quadratic edges and four linear edges, while the 20-node element is fully quadratic. Note that the Serendipity elements do not have face nodes or body nodes as do the Lagrangian elements. The 24-node element is similar to the 16-node element in that it has four linear edges. The other eight edges, however, are cubic. The 32-node element is fully cubic. Experience with the 16- and 24-node elements has shown that large inplane-to-out-of-plane aspect ratios can be handled by orienting the element such that the linear edges are in the through-the-thickness direction of the laminate. This is important, since laminates typically are many orders of magnitude larger in planform than in thickness. The element stiffness matrices are evaluated by Gaussian quadrature at a number of Gauss points which is user selectable.

MODELING ISSUES

- Element selection
- Number of elements per ply or fiber orientation
- Aspect ratio
- Location of 2-D/3-D transition region

MODELING ISSUES

A number of modeling issues arise in the 3-D analysis of laminates. Some of these issues are unique to composites, while others are always encountered in finite element analysis. They include:

1. Decision of which element to use. The Testbed, in its current form, offers the choice of either hybrid stress or hybrid strain formulation elements in addition to the displacement elements which have been introduced. The choice of elements from the library of a general purpose program has been, is, and will probably continue to be at the discretion of the user.

2. If the laminate is to be modeled such that the interlaminar stresses are determined, each ply, or at least each group of plies with the same orientation, must be subdivided into some number of elements through its thickness. The number of elements through the thickness is of concern. The proper number of elements through the thickness is highly dependent on the order of the element.

3. Since the planar dimensions of a laminate are typically several orders of magnitude greater than the thickness dimension, element aspect ratio will undoubtedly be a problem when conducting 3-D analyses.

4. The location of the global/local interface must be determined. Previous investigators have reported differences in stress prediction due to different interface locations (e.g. [6]). If global/local analysis is to become routine, specific guidelines and techniques for location of an optimum interface position must be determined.

ELEMENT SELECTION

- Based on problem to be solved
- Curved surfaces not easily modeled with linear elements
- Doubly curved surfaces cannot be easily modeled with elements with any linear edges
- Requirements for elements with highly curved faces may require Lagrangian elements
- Special requirements imposed by woven or braided composites
- Thin and thick sections may require different elements due to differences in acceptable aspect ratio

ELEMENT SELECTION

Element selection is based on a number of factors. They include:

1. The problem which is being solved. The choice of elements for a particular analysis usually depends on the nature of the problem, including geometry, material characteristics, and loading. A grid which might be completely acceptable for analysis of a structure or component under one loading condition might be unacceptable for a different loading. This choice is normally based on the experience of the analyst.

2. Some problem characteristics preclude the use of certain elements for a particular analysis. For example, it is not easy to model accurately a curved surface with elements which have linear edges. Approximation of curves by a series of straight lines may sometimes be acceptable, but this is often not the case. The use of elements which have some linear edges may be possible.

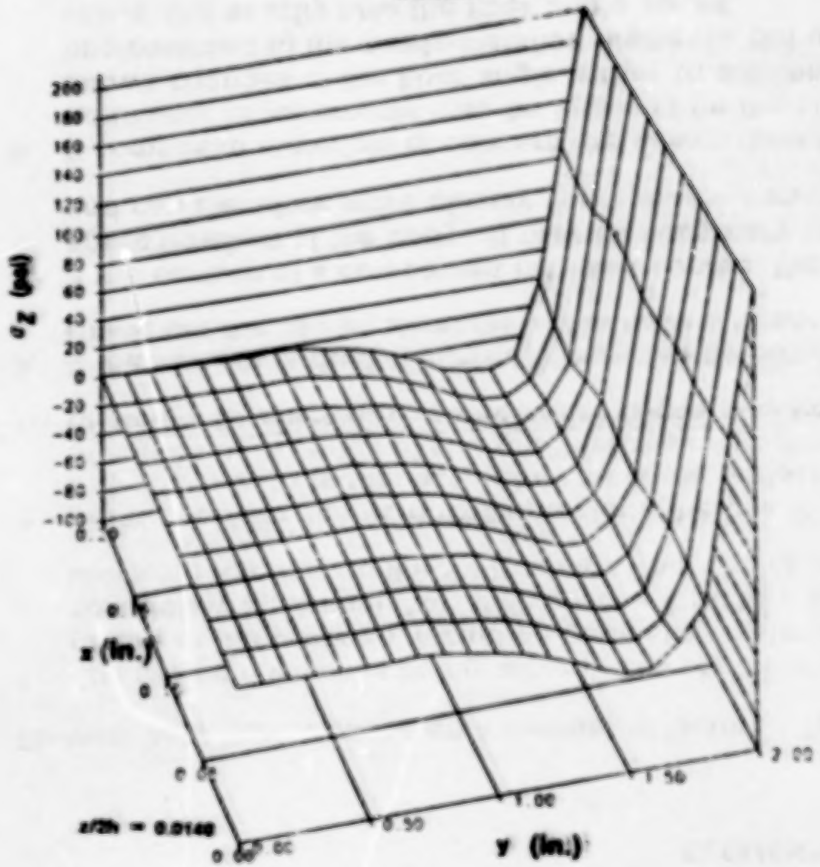
3. If the surface is doubly curved, it may not be possible to use elements which have any linear edges, thus ruling out the use of elements which have a linear geometric shape function on any edge.

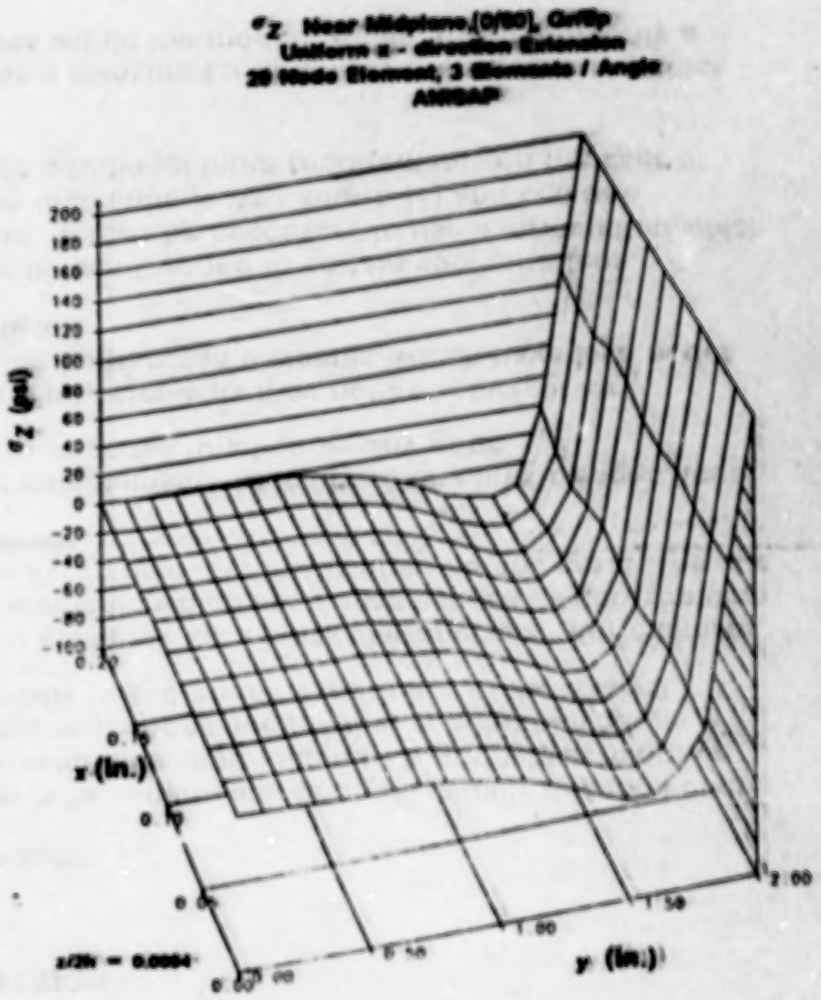
4. If the surface of a component is highly curved, the Serendipity elements may not be sufficient for approximation of the body. It may be necessary to use the Lagrangian elements, which have face nodes and can therefore more closely approximate curved surfaces.

5. If composite materials of concern are woven, braided, or filament wound in certain configurations, additional requirements may be imposed on the elements. It may be necessary to use a formulation which allows changes in the fiber angle within an element. This technique is well known [7] and can be implemented in the newly-installed elements, but imposes additional input requirements on the user of specifying exactly how the fiber angle varies.

6. If the geometry of a structure changes drastically, such as a transition from a very thin section to a thick section, it may be necessary to use a mix of element types within the model. Again, this is primarily a decision forced on the analyst, based on experience.

STUDY OF NUMBER OF ELEMENTS PER PLY





STUDY OF NUMBER OF ELEMENTS PER PLY

The reason that a number of elements must be used through the thickness of a particular ply or group of plies at the same angle is the existence of large gradients of stress within the laminate. Some determination must be made of the proper number of elements to use through the thickness, based on a trade-off between accuracy in modeling the gradients and problem size, which becomes substantially larger for every layer of elements which is added. Experience has shown that three linear elements or two quadratic elements through the thickness of each ply angle is sufficient. This is supported by experimental results using Moire interferometry [4]. This slide shows the results obtained for interlaminar normal stress (σ_z) near the midplane for uniform extension in the x-direction of a cross-ply laminate. Note that a smooth distribution is obtained for both two and three elements through the thickness of each ply group. The values are slightly different since the Gauss points for the three elements per layer are slightly closer to the midplane than in the two elements per layer. Examination of the through-the-thickness distribution, however, indicates that either two or three elements per layer gives essentially the same distribution, while the two elements per layer results in a considerably more economical analysis. These results were generated using the finite element program ANISAP [8].

ELEMENT ASPECT RATIO

- Laminates typically orders of magnitude greater in planar dimensions than in thickness
- Aspect ratio difficulties arise immediately when using 3-D elements
- Acceptable aspect ratios different for various loading conditions
- Element failure is due to ill-conditioning of stiffness matrix

ELEMENT ASPECT RATIO

It is well known (e.g. [9], [10]) that problems arise when element aspect ratios are not within certain bounds. Ideally, there should be approximately equal strain in all directions for displacement formulation elements [11]. Without a priori knowledge of the solution, however, it is impossible to satisfy that condition. Furthermore, in laminates the planar dimensions of most parts of practical interest are orders of magnitude greater than the thickness. Since a number of elements must be used through the thickness when conducting 3-D analyses, either the element aspect ratio or the number of elements becomes excessive almost immediately. In production analyses, large numbers of elements, with associated large solution time (and cost) are generally unacceptable. Large element aspect ratios result in erroneous results in many cases, most likely due to numerical ill-conditioning of the element stiffness matrix. There is also the potential for an element to become too stiff in one component or the other. The problem is also complicated by the anisotropy of the composite, since for common aerospace materials the ratio of fiber direction modulus (E_1) to transverse direction modulus (E_2) may range from 10 to 100. Hence, optimum inplane aspect ratios may be different than would be expected, due to large differences in strain.

EVALUATION OF EXISTING TESTBED CAPABILITIES

- **S81 (8-node brick) Element**
  - **Uniform extension correct for unidirectional and clustered cross-ply laminates with material and element axes coincident (0 and 90 degree plies)**
  - **Problems when material axes not coincident with element axes (45 degree plies) determined to be LAU problem, now fixed.**
  - **Acceptable in-plane-to-out-of-plane aspect ratio problem dependent.**
- **E43 (plate/shell) element works**

EVALUATION OF EXISTING TESTBED CAPABILITIES

In addition to installing new 3-D elements in the CSM Testbed, work has been done to evaluate the original capabilities, specifically the 3-D brick assumed stress element (S81). The S81 element originally implemented in the Testbed works properly for composite materials so long as the element faces are parallel. The 3-D capabilities of the Laminate Analysis Utility (LAU) were used to generate the property tables for composites. Uniform extension in the x-direction was analyzed for cross-ply and angle-ply symmetric laminates. Original results indicated a problem with the angle-ply laminates. This was later determined to be due to an error in LAU, which has now been corrected. The problem of errors with elements with nonparallel faces has been fixed by NASA Langley CSM personnel. As expected, acceptable S81 element aspect ratios have been determined to be problem dependent. Original results for cross-ply laminates indicated a maximum in-plane:thickness ratio of 6:1 before numerical difficulties resulted in a fatal underflow, but other analyses have been run with much larger aspect ratios, with good results.

Evaluation of the E43 element, again using LAU to generate the property tables, has shown this element to be reliable. These analyses were, however, conducted using rectangular elements with reasonable aspect ratios.

GLOBAL/LOCAL ANALYSIS TECHNIQUES

- **Global region**
  - **Variational (e.g. Rayleigh-Ritz) solutions**
  - **Analytical solutions**
  - **Coarse finite element models**
- **Local region**
  - **Finer grid of 2-D or 3-D elements**
  - **Substructuring**
  - **Special elements**
  - **Displacement and/or slope matching constraints**

GLOBAL/LOCAL ANALYSIS TECHNIQUES

One method of overcoming the previously discussed problems of either large aspect ratios or large problem size is to perform some sort of global/local analysis. There are several techniques for such analyses, both in the global and local regions. These include:

- Global region
  - Variational solutions such as Rayleigh-Ritz have been widely used for solution of plate problems and are a viable choice for the global analysis of plate-type structures. These solutions are well known and trusted.
  - Other analytical solutions may be available for other types of problems. Choice of such models must be made on a case-by-case basis. However, the solution must be valid for the problem of concern, excluding the local effect, which must also be determined to be so local that it does not affect the accuracy of the chosen global model.
  - Coarse finite element models which are accurate for the global response are often chosen for the global region. The use of coarse 2-D models for 2-D problems with local 2-D effects is common. Often a coarse 2-D model, followed by a finer 2-D model, to be followed by a local 3-D model may be used.
- Local region
  - Appropriate choices for a local model may simply be finer discretization of a 2-D or possibly a 3-D finite element model. In that case, the displacements, rotations, etc. from the global model may be applied as boundary conditions on the fine model. One must be careful, however, that the interface between the global and local regions chosen is such that the global model is accurate, that is, the interface must be outside of the locally-affected region.
  - Substructuring, or the use of "superelements," which would be followed by static condensation and then solution of the equations, may be used. This option might be particularly attractive if the software being used offers ready use of this technique.
  - The use of special elements, often with many degrees of freedom, or special "transition" elements is also an alternative. This option, however, requires the software to provide these capabilities, often requiring the developers to have correctly anticipated the classes of problems which will be of interest to users.

RESULTS

- Benchmark of 20-node Experimental Element in Testbed
  - Two rectangular parallelpiped elements with angled boundary face
    - Face angle to 50 degrees gives uniform stress state for uniform force loading
- Uniform extension of cross-ply laminates
- [0/90]s, [90/0]s, [0/0/90/90]s, and [90/0/90/0]s Gr/Ep Laminates run on CSM Testbed and ANISAP using 20-node displacement element

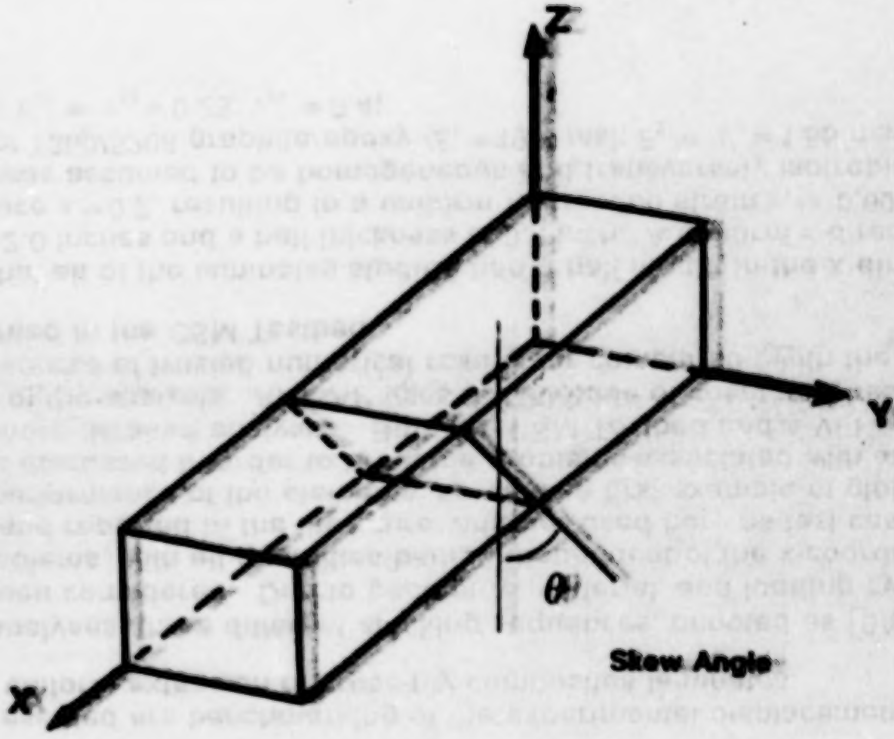
RESULTS

The results to be presented are benchmarking of the experimental displacement elements and for some problems involving uniform extension of cross-ply composites laminates.

For the composite analyses, three different stacking sequences, denoted as $[0/90]_s$, $[90/0]_s$, $[0/0/90/90]_s$, and $[0/90/0/90]_s$ have been considered. Due to geometric, material, and loading symmetry, these are actually two-dimensional problems, with all quantities being independent of the x-coordinate. They have been thoroughly studied and reported in the literature, and are used here as test cases with well known results in order to check the performance of the elements, serve as a first example of global/local analysis, and to yield results which can be discussed in order to bring the problems associated with analysis of composites into focus before going on to more detailed analyses. Both the CSM Testbed and a VPI program known as ANISAP [8] were used for parts of the analysis. ANISAP uses the 20-node element and has been well checked out, and was thus used as a source of trusted numerical results for comparison with the Experimental Elements which have been implemented in the CSM Testbed.

It should be noted that all of the laminates studied had a half length in the x-direction of 0.2 inch, a half width in the y-direction of 2.0 inches and a half thickness of 0.4 inch. A uniform x-direction displacement of 0.001 inch was applied at the face $x = 0.2$, resulting in a uniform x-direction strain $\varepsilon_x = 0.005$. For the analyses, the composite material was assumed to be homogeneous and transversely isotropic, with the properties approximately that of T300/5208 graphite/epoxy ($E_1 = 19.2$ msi; $E_2 = E_3 = 1.56$ msi; $G_{12} = G_{13} = 0.83$ msi; $G_{23} = 0.5E_2/(1 + \nu_{23})$; $\nu_{12} = \nu_{13} = 0.23$; $\nu_{23} = 0.4$).

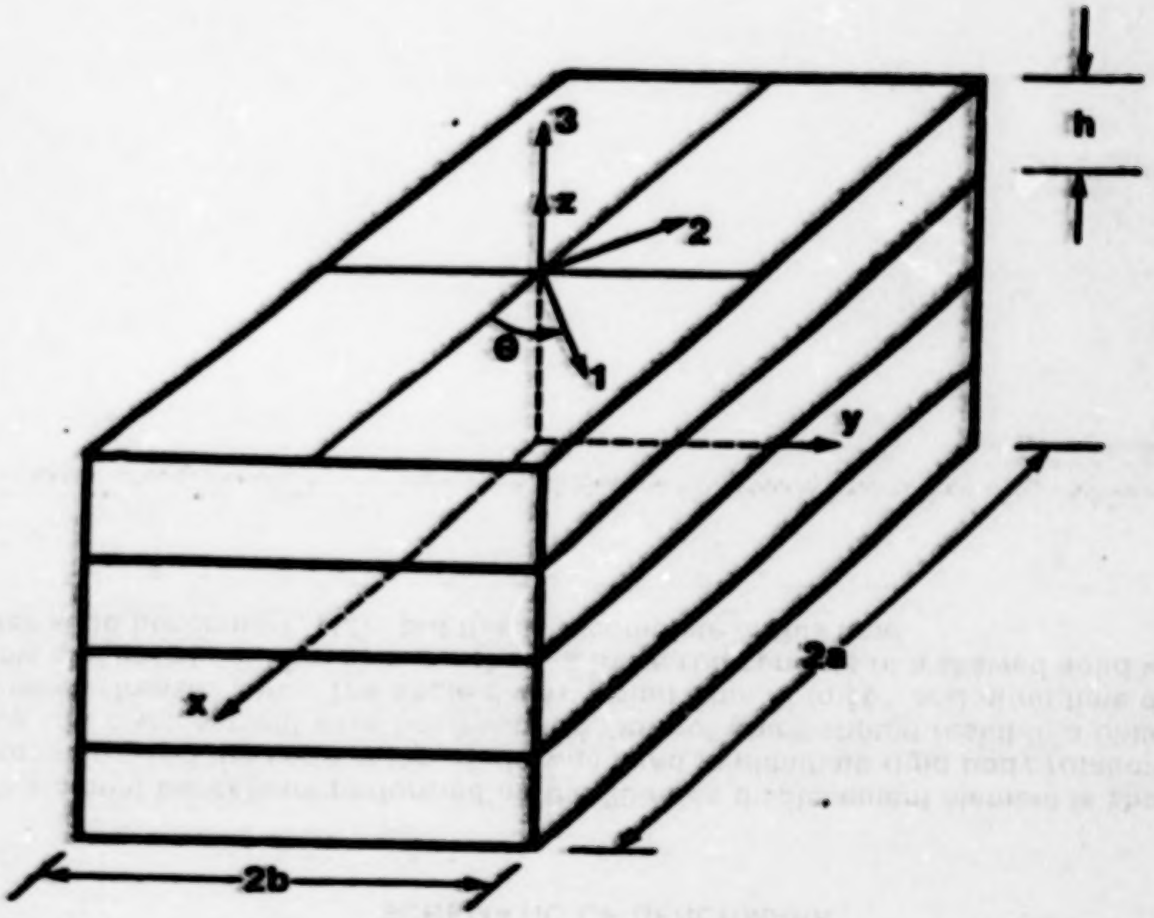
SCHEMATIC OF BENCHMARK



SCHEMATIC OF BENCHMARK

The simple, two element benchmark performed on the 20-node displacement element is shown. The y-z plane is a plane of symmetry, and the node at the origin was fixed to eliminate rigid body rotation. Both uniform x-direction force and displacement have been applied, both of which should result in a uniform x-direction stress, with all other stresses zero. The angle θ was varied from 0° to 50°, and at no time did the stress distribution show any variation. The next benchmark to be run consists of a skewed solid within a cube (MacNeal-Harder solid benchmark, [12]), but this is incomplete at this time.

SCHEMATIC OF LAMINATE

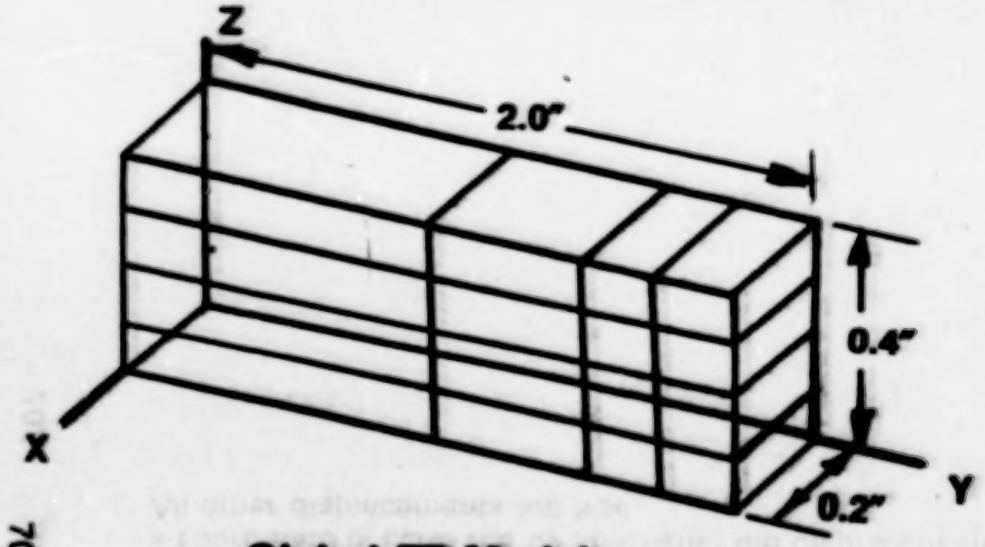


SCHEMATIC OF LAMINATE

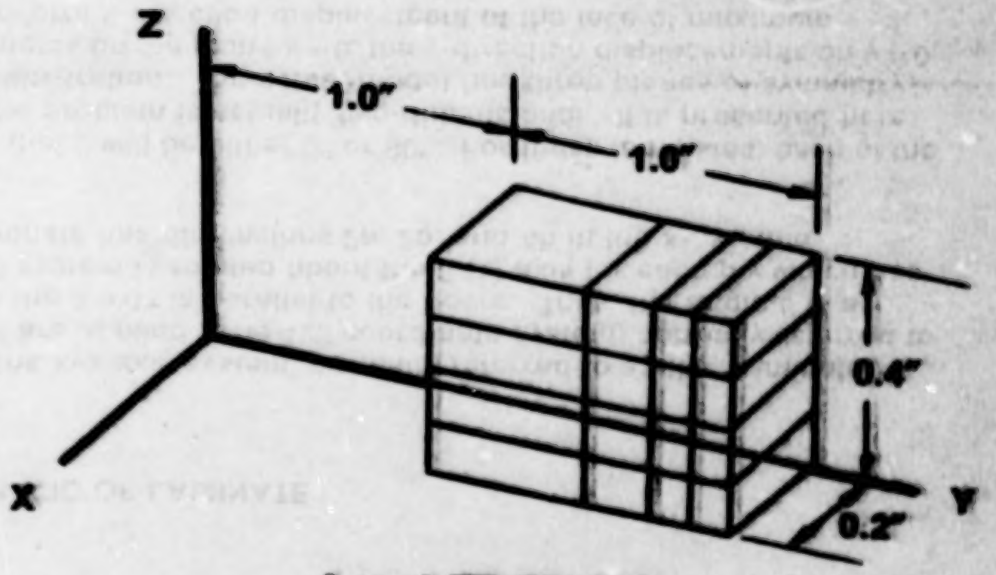
The slide shows a general laminate geometry. The xyz axis system, normally referred to as the "laminate coordinate system," is the system in which loads are applied. The 123 coordinate system, normally referred to as the "material coordinate system," is such that the 1-axis is parallel to the fibers. Thus, the angle θ is a measure of how the fibers are oriented. The 123 system is rotated about the 3 (z) axis for each ply which has a fiber angle (θ) which may be nonzero. The laminate has dimensions 2a, 2b, and 4h in the x-, y-, and z-directions, respectively.

Note that for the laminates to be discussed here that θ will be either 0° or 90°. For these laminates, each of the coordinate planes is a plane of symmetry, and the problem is actually two-dimensional. It is presented here as a three-dimensional analysis for purposes of illustration. The actual model has three planes of symmetry, achieved by restraining the x-direction displacements on the plane $x = 0$, the y-direction displacements on $y = 0$, and the z-direction displacements on $z = 0$. A uniform x-direction displacement of the face of maximum x-coordinate is achieved by specifying the displacement of all nodes on that face to have the desired value. All other displacements are free.

FE MODELS USED FOR GLOBAL/LOCAL ANALYSIS



Global FE Model



Local FE Model

FE MODELS USED FOR GLOBAL/LOCAL ANALYSIS

The finite element grids shown were used for global and local analyses of the laminates. Each grid consists of two elements per ply orientation, four elements in the y-direction, and one element in the x-direction. Only the 20-node displacement elements were used in the laminate analyses.

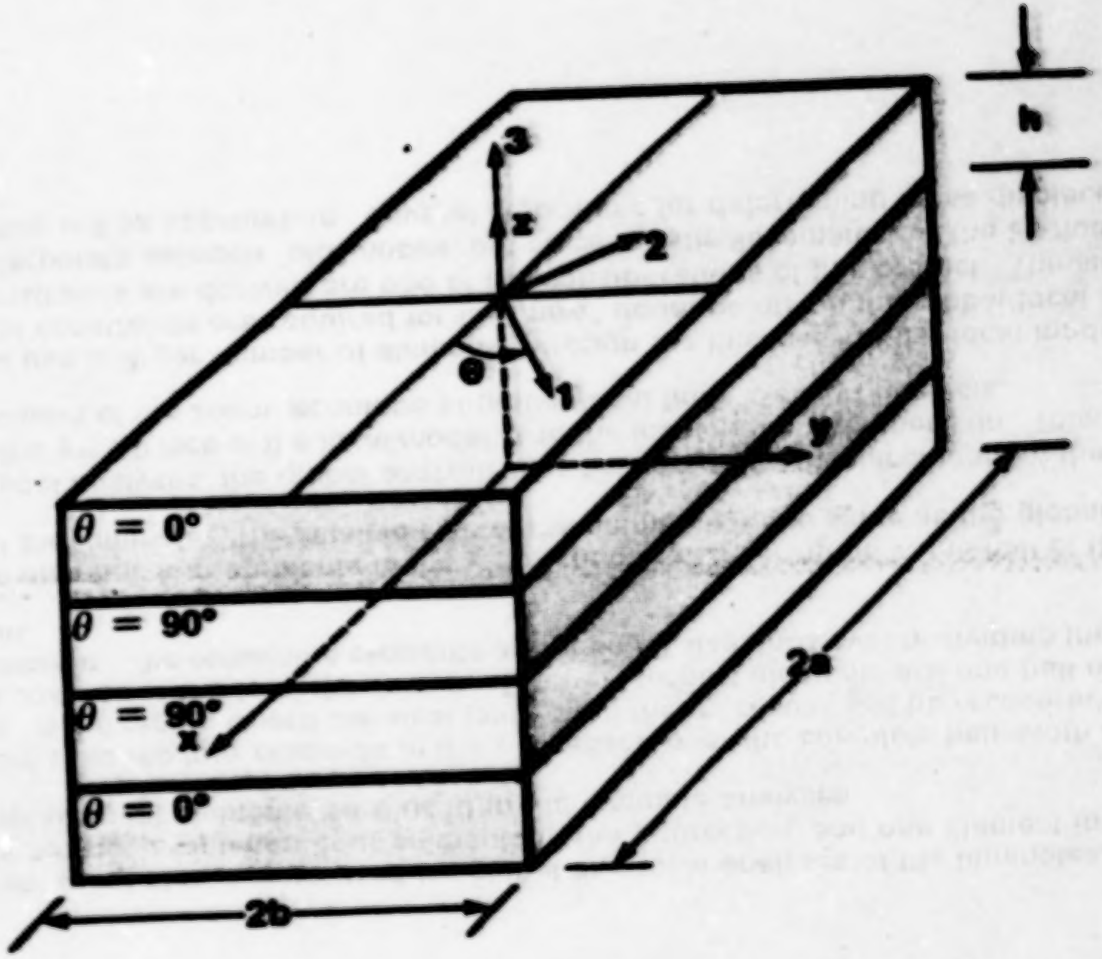
The global model uses the four elements in the y-direction over the complete half width of the laminate. The largest element, in the region where classical lamination theory results will be recovered and the gradients are the smallest, is one-half of the total dimension. The next largest elements are one-half of the remaining y-direction dimension. The remaining elements are of equal size, thus evenly dividing the remaining one-fourth of the dimension.

The local model also has four elements in the y-direction, but covering only one-half of the y-dimension of the modeled part of the laminate. The relative size of the elements is the same as the global model.

For the global/local analyses, the global analysis was first run. The displacements at the plane $y = 1.0$ were then applied to the $y = 1.0$ face of the local model, and the local model was then run. This corresponds to a very simplified application of the zoom technique of global/local finite element analysis.

If it is desired to use a larger number of elements through the thickness in the local model than in the global model, additional constraints are required for the "new" nodal points on the global/local interface. The details of how these constraints are derived are one of the considerations of this project. The simplest method would be to linearly interpolate between "old" nodes, but since the displacements for the 20-node element are quadratic, that may not be appropriate. Several algorithms for determining these displacements are discussed in Reference 13.

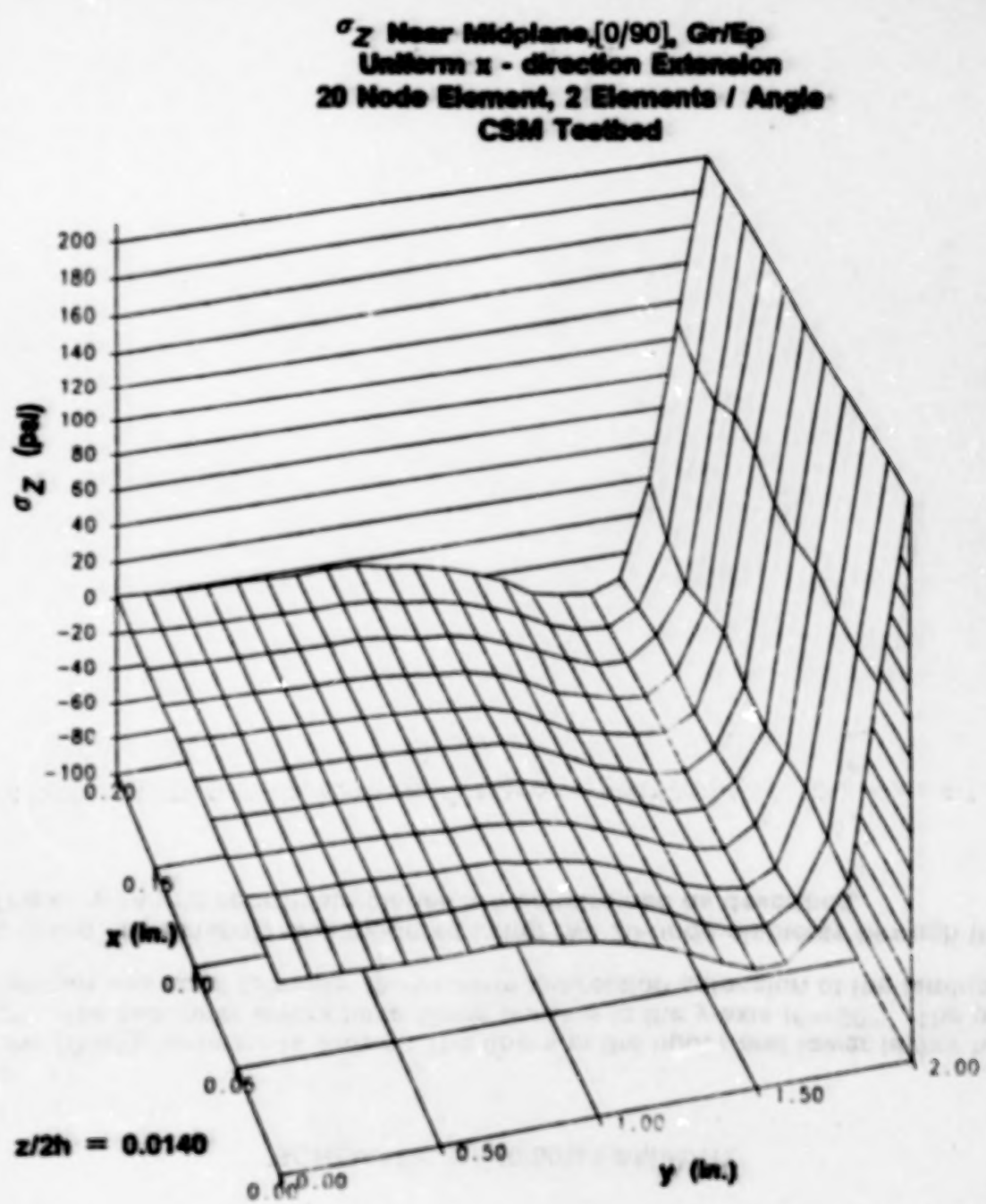
SCHEMATIC OF [0/90]s LAMINATE



SCHEMATIC OF [0/90]s LAMINATE

The schematic of the $[0/90]_s$ laminate is shown. The fibers in the upper and lower layers have fibers parallel to the x-axis ($\theta = 0°$). The two inner layers have fibers parallel to the y-axis ($\theta = 90°$). The global finite element model previously shown was used to model the uniform x-direction extension of the laminate.

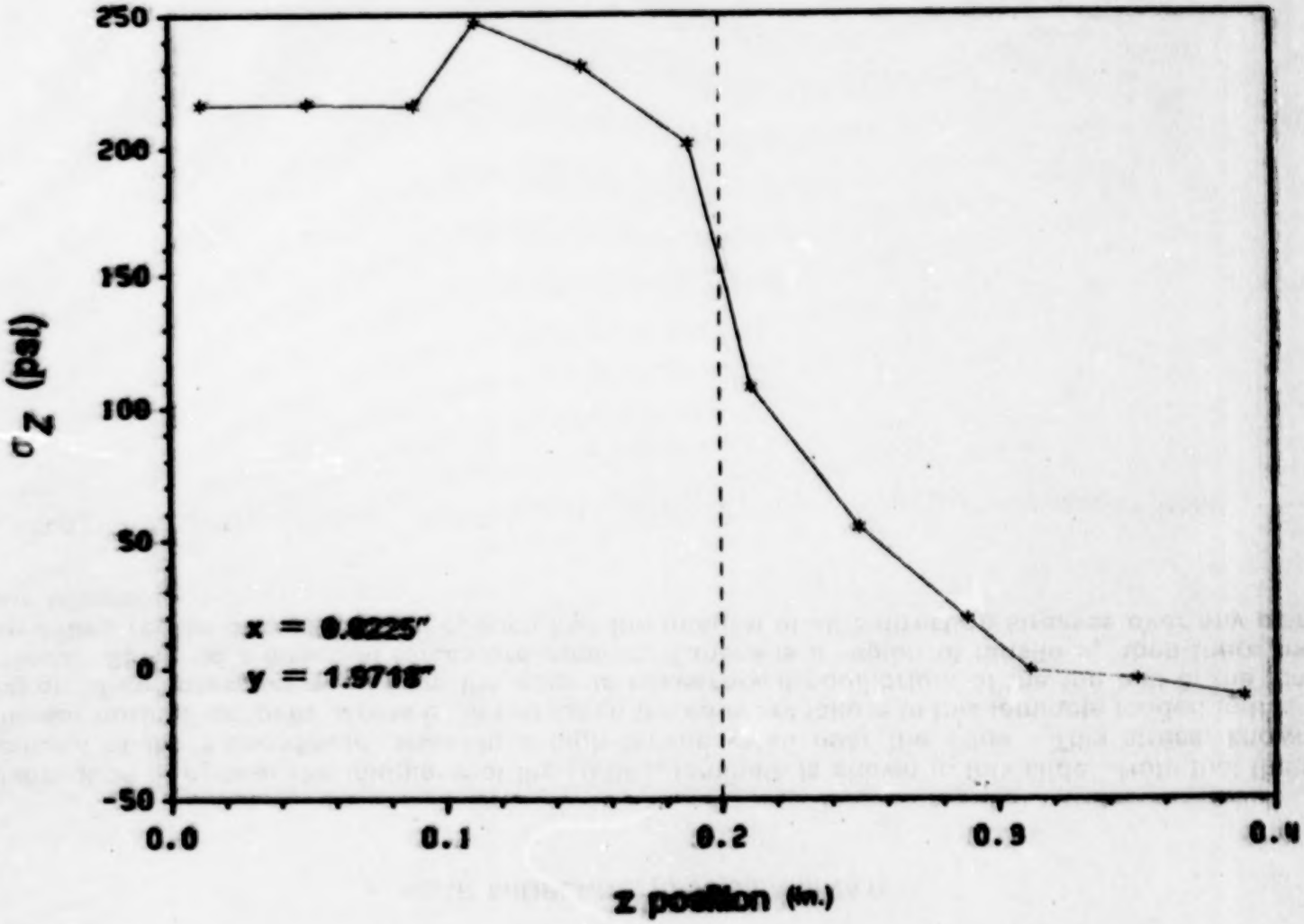
As previously discussed, the laminate was modeled using two 20-node elements through the thickness of each ply orientation. The x-, y-, and z-coordinate planes are constrained as described.


σz Near Midplane,[0/90], Gr/Ep
Uniform x - direction Extension
20 Node Element, 2 Elements / Angle
CSM Testbed
σz (psi)
200
180
160
140
120
100
80
60
40
20
0
-20
-40
-60
-80
-100
x (in.)
0.20
0.00
y (in.)
0.00
0.50
1.00
1.50
2.00
z/2h = 0.0140

σ_z NEAR MIDPLANE, [0/90]s LAMINATE

The distribution of σ_z near the midplane of the $[0/90]_s$ laminate is shown in this slide. Note that the stress is independent of the x-coordinate, showing a high tensile value near the edge. This stress, known as the interlaminar normal, or "peel" stress is, in fact, often the cause of failure in this laminate loaded in this manner. The region of compression away from the edge is necessary if equilibrium of the top half of the laminate is considered. Since no z-direction forces are applied, if there is a region of tensile σ_z, then there must be a corresponding region of compressive σ_z such that the integral of all z-direction stresses over any plane in the laminate is zero.

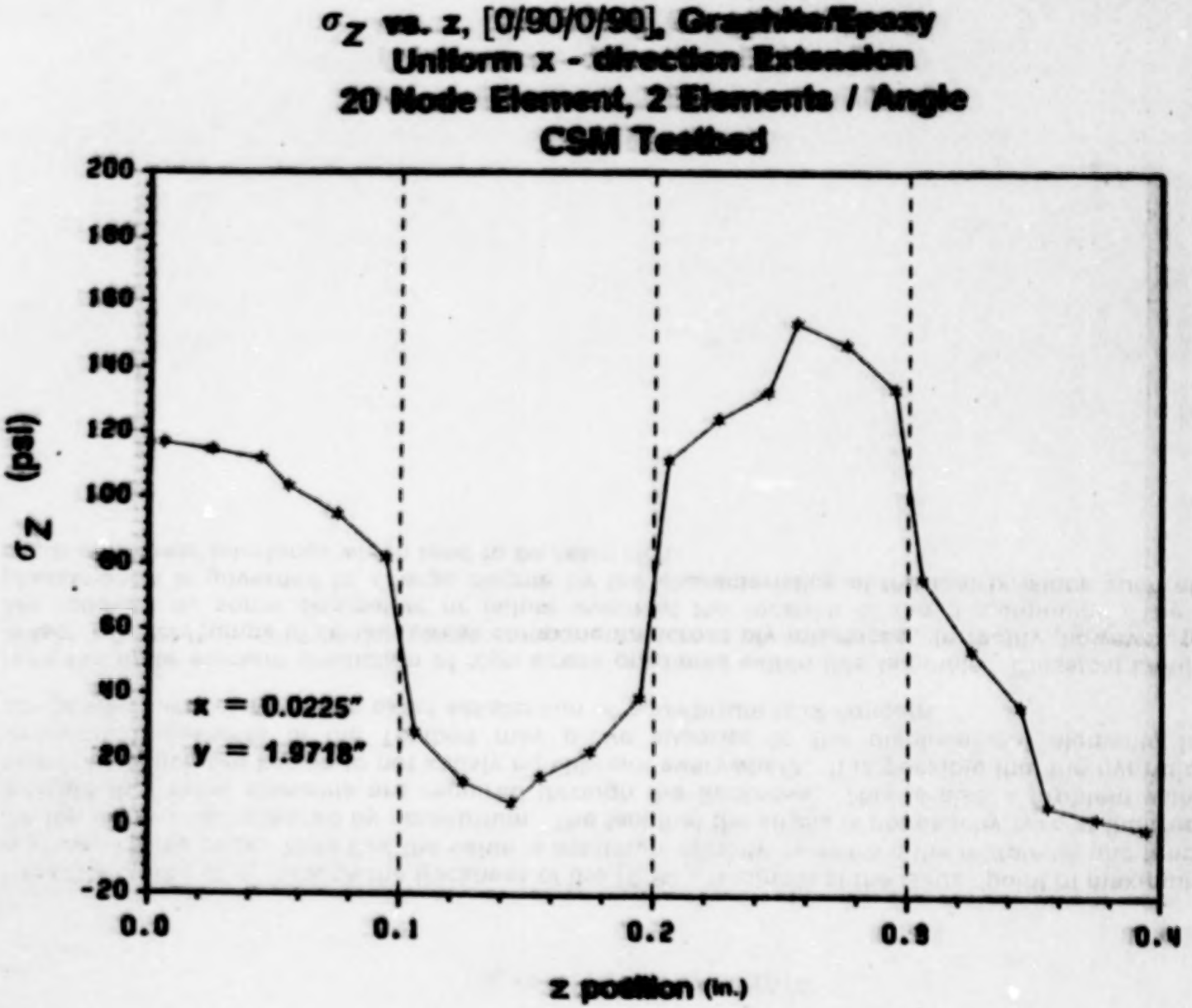
σ_z vs. z, [0/90]$_s$ Graphite/Epoxy
Uniform x - direction Extension
20 Node Element, 2 Elements / Angle
ANSAP



σ_z vs. z, [0/90]s LAMINATE

The distribution of σ_z through the thickness of the $[0/90]_s$ laminate at the Gauss point of maximum y-coordinate is shown in this slide. Note that the value is maximum slightly away from the midplane, and tends to zero near the top surface, as required by equilibrium. The fact that the stress is not exactly zero at the free surfaces may indicate that more elements are required through the thickness. This is also a problem with displacement elements, which are known to not satisfy equilibrium everywhere. It is possible that the hybrid stress or strain formulation elements of the Testbed may prove superior to the displacement elements for analysis of composites, especially where exact satisfaction of equilibrium is of concern.

Note the finite element prediction of high stress gradients within this laminate. Classical Lamination Theory, in fact, predicts jumps in certain stress components across ply interfaces. In reality, however, jumps in stress are relieved by some dissipative or failure event at the location of the discontinuity. The nature of this phenomenon is governed to a large degree by the characteristics of the matrix, since such effects normally occur at or near interfaces which tend to be resin rich.


σz vs. z, [0/90/0/90], Graphite/Epoxy
Uniform x - direction Extension
20 Node Element, 2 Elements / Angle
CSM Testbed
σz (psi)
200
180
160
140
120
100
80
60
40
20
0
-20
x = 0.0225"
y = 1.9718"
0.0
0.1
0.2
0.3
0.4
z position (in.)

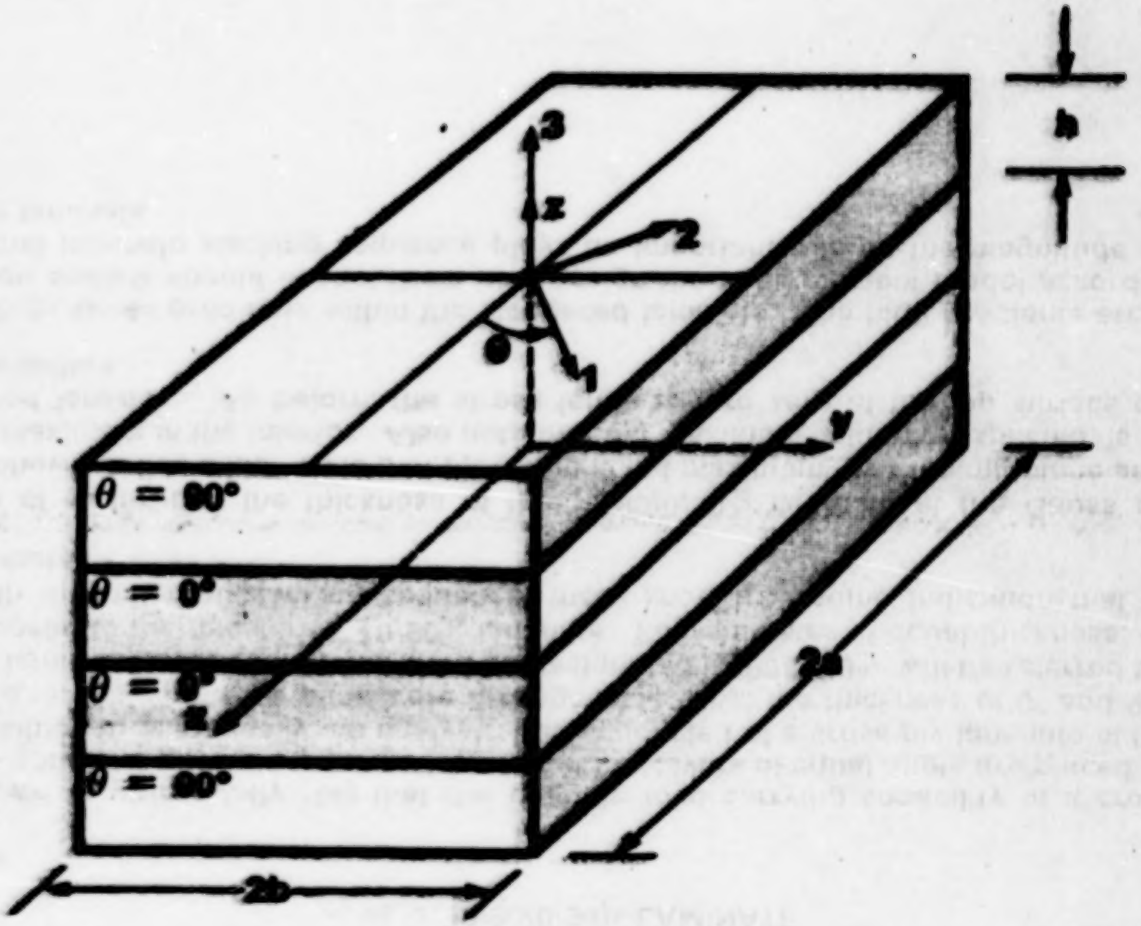
σ_z vs. z, [0/90/0/90]s LAMINATE

It has been shown experimentally [14] that the ultimate load carrying capability of a cross-ply laminate is increased if the individual plies are arranged such that few layers of equal angle are placed together. In order to examine this situation numerically, an analysis was conducted of a cross-ply laminate of the same thickness as the previous one, but where the plies were arranged such that the thickness of 0° and 90° ply groups was cut in half. This laminate, which could therefore be designated $[0/90/0/90]_s$, will be referred to as a "staggered" laminate, as opposed to the "clustered" $[0/90]_s$ laminate. For laminates of equal thickness, same material and loading, but with the different stacking sequence, there should be some indication that σ_z is lower in the "staggered" laminate.

The distribution of σ_z through the thickness of the $[0/90/0/90]_s$ laminate at the Gauss point of maximum y-coordinate is shown in this slide. Note that the value is not maximum near the midplane as with the clustered laminate, but is maximum in the interior. Also note that the maximum value, as expected, is substantially lower than the clustered laminate. As before, the stress tends toward zero at the top surface of the laminate, as required by equilibrium.

There are also high stress gradients within the staggered laminate. The high gradients are to be expected in laminates, and an analyst should expect them and design the finite element model accordingly. One should also be aware that laminate stacking sequence plays an important role on the magnitude and distribution of stress within the laminate.

SCHEMATIC OF [90/0]s LAMINATE



SCHEMATIC OF [90/0]s LAMINATE

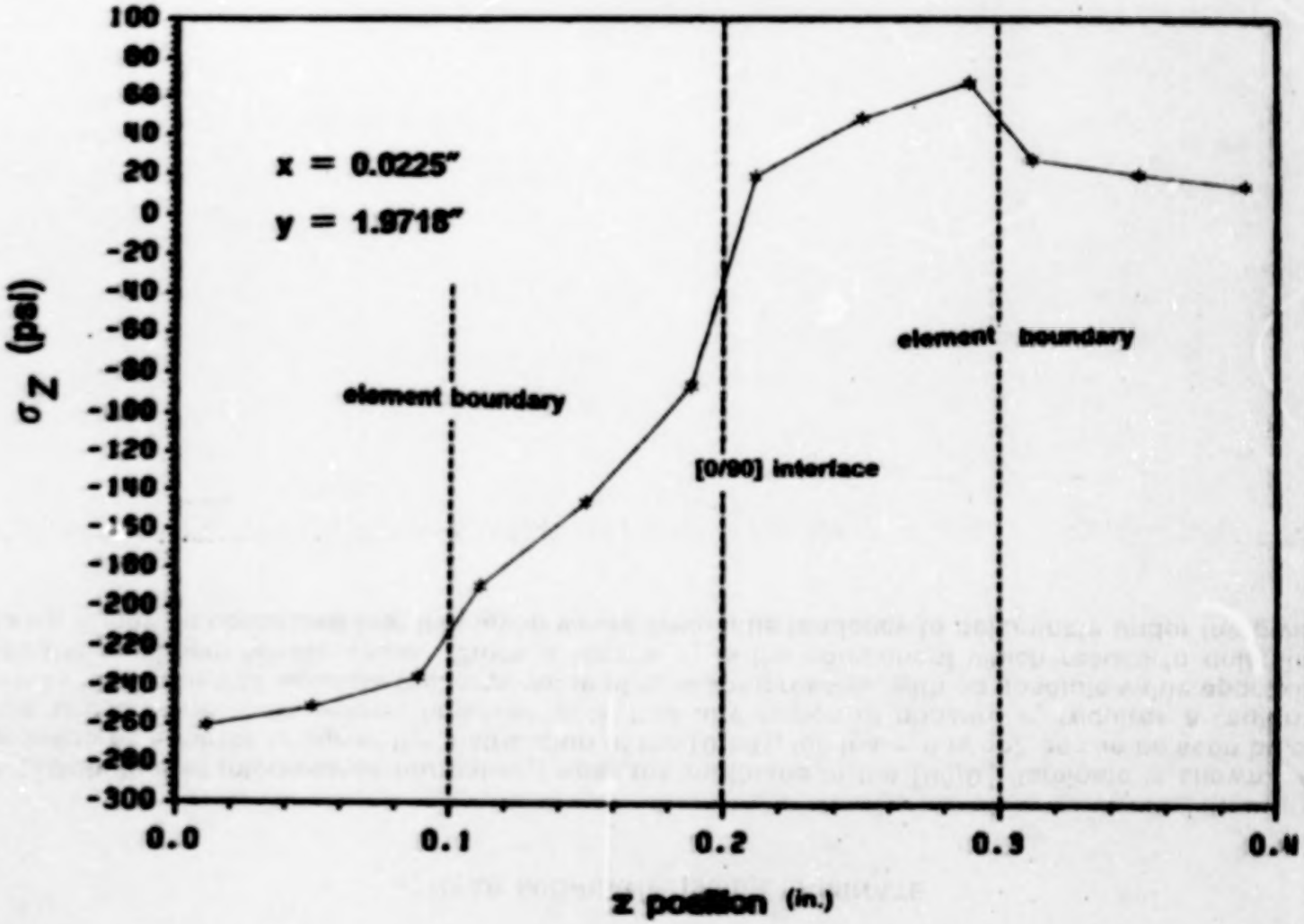
The schematic of the $[90/0]_s$ laminate is shown. The fibers in the upper and lower layers have fibers parallel to the y-axis ($\theta = 90°$). The two inner layers have fibers parallel to the x-axis ($\theta = 0°$). The global finite element model previously shown was used to model the uniform x-direction extension of the laminate.

ORIGINAL PAGE IS
OF POOR QUALITY

σ_z NEAR MIDPLANE, [90/0]s LAMINATE

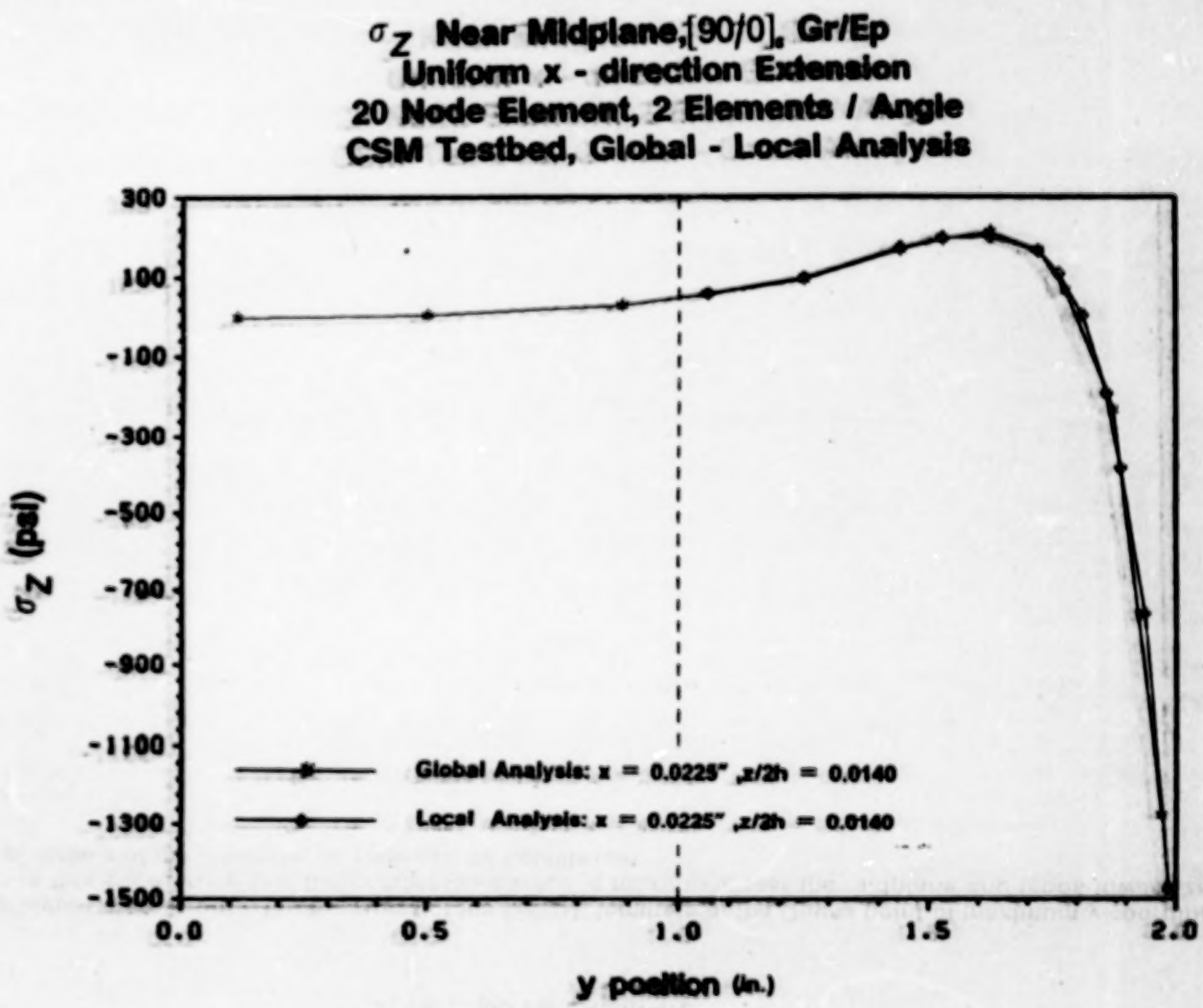
The distribution of interlaminar normal (σ_z) near the midplane of the $[90/0]_s$ laminate is shown. Although it appears to be a mirror image of the distribution of the $[0/90]_s$ laminate, it is not, as can be seen by comparison of the stress values. As before, however, note that any region of nonzero σ_z requires a region of σ_z with opposite sign. For this laminate the large value of σ_z is compressive, with an absolute value approximately five times the maximum tensile value. Since a tensile σ_z is the component which results in delamination type failures, it may be concluded that this layup would have little tendency to delaminate under the given load.

σ_z vs. z, [90/0]$_s$ Graphite/Epoxy
Uniform x - direction Extension
20 Node Element, 2 Elements / Angle
ANISAP



σ_z vs. z, [90/0]s LAMINATE

The distribution of σ_z through the thickness of the $[90/0]_s$ laminate at the Gauss point of maximum y-coordinate is shown in this slide. Note that the compressive value is maximum near the midplane and tends toward zero at the top surface of the laminate, as required by equilibrium.

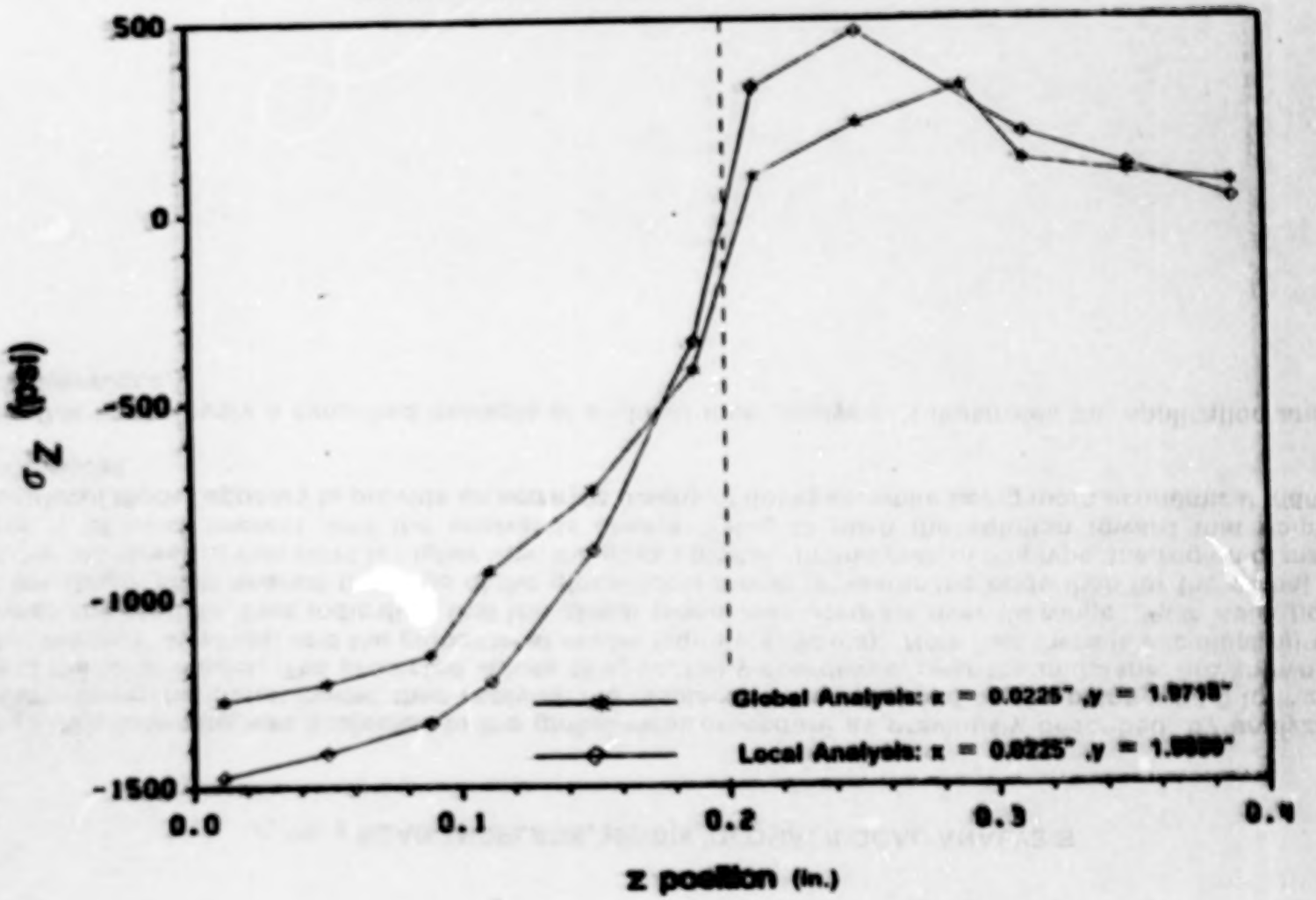

σz Near Midplane,[90/0]s Gr/Ep
Uniform x - direction Extension
20 Node Element, 2 Elements / Angle
CSM Testbed, Global - Local Analysis
300
100
-100
-300
-500
-700
-900
-1100
-1300
-1500
σz (psi)
0.0
0.5
1.0
1.5
2.0
y position (in.)
Global Analysis: x = 0.0225" ,z/2h = 0.0140
Local Analysis: x = 0.0225" ,z/2h = 0.0140

σ_z vs. y NEAR MIDPLANE, [90/0]s, GLOBAL/LOCAL ANALYSIS

The $[90/0]_s$ laminate was subjected to the global/local procedure as previously described, by analyzing the laminate using the global model, then applying the displacements calculated on the plane y = 1.0 to the y = 1.0 face of the local model. The predicted values of σ_z vs. the y-coordinate, near the midplane, are shown for the global model (* symbols) and the global/local model (square symbols). Note that there is excellent agreement between the models, thus indicating that the global model was accurate over its range. Also note, however, that the Gauss point nearest the edge of the global/local model is nearer the edge than for the global model, and that the stress at that point is higher than the global model. In analyses of this type, the region of maximum stress is of most interest, and the analyst is always trying to push the solution toward that point. The global/local model appears to provide an accurate means of doing so, while being more economical than a finer global model.

While this is obviously a simplified example of a global/local analysis, it illustrates the application and utility of the technique.

σ_Z vs. z, [90/0], Graphite/Epoxy
Uniform x - direction Extension
20 Node Element, 2 Elements / Angle
CSM Testbed, Global - Local Analysis



σ_z vs. z, [90/0]s LAMINATE, GLOBAL/LOCAL ANALYSIS

The distribution of σ_z through the thickness of the $[90/0]_s$ laminate at the Gauss point of maximum y-coordinate for the global and global/local models is shown in this slide. Note that the global/local model yields a higher compressive stress near the edge, as expected. It also appears that the tensile stress is higher than predicted by the global model, and at a slightly different location in the laminate. This may not be a disagreement between the models, however, since the Gauss points at which the global/local stress is plotted is closer to the edge than for the global model.

CONCLUSIONS

- Installing Experimental Elements in CSM Testbed is straightforward once you know the rules
- Once installed in the Testbed, new elements are readily benchmarked
- As previously shown experimentally, laminates with "staggered" plies (e.g., $[0/90/0/90]_s$) exhibit substantially lower interlaminar stresses than laminates with "clustered" plies (e.g., $[0/0/90/90]_s$)
- First runs of 3-D Global/Local analysis using 20-node displacement elements in Testbed were successful. Results compare with proven software

CONCLUSIONS

A number of conclusions, both on the use of the CSM Testbed and the mechanics of laminated composites, can be drawn from the work conducted to date. They include:

1. Installing new elements in the CSM Testbed using the Experimental Element Capability is straightforward once you know all of the rules. The documentation on this capability is somewhat fragmented and may be incomplete. It appears that the Generic Element Processor, which will be in the new release of the Testbed, will be of great utility in efforts such as the present study. It provides an even easier means of installing new elements for experimentation and use, as well as providing geometric nonlinearity via the corotational technique. It appears that in an educational environment, where the time a student is actually working on a degree may be the limiting factor, this capability will allow researchers to readily implement new elements and then spend more time testing and using the elements rather than divert valuable time toward the development of complete new software implementations before testing and use can begin. This is very valuable to those of us who are more interested in developing new capabilities and then immediately using them to study the problems which are of real interest to us, as opposed to those who are more interested in developing the new capabilities and then turning them over to others for use.

2. Once installed in the Testbed, new elements are readily checked out, benchmarked, and ready to use. There is no additional time required to test the validity of assembly routines, solvers, etc., since those are provided by the Testbed.

3. It has been shown that cross-ply symmetric laminates with "staggered" plies have lower interlaminar normal stresses due to uniform extension than do laminates with "clustered" plies. This confirms experimentally observed behavior.

4. Simple global/local finite element analysis using coarse and fine grids of 20-node displacement elements yielded good results. While the global model appears to be accurate, use of the global/local model enables prediction of stresses nearer the edge than with the global model.

PLANS

- Use "zoom" technique for local/global analyses
- Solve problem of pure bending of composite plate with central circular hole using CLT and/or E43 elements for global solution and 3-D FEA for local solution
- Add stiffener(s) with at least one interrupted by hole
- Consider stress distributions around stiffener/hole/plate junctures
- Determine guidelines for location of local/global interface, semi-automatic if possible

PLANS

The zoom [13] technique will be used for global/local analysis. This technique offers a straightforward method which allows all analyses to be done with the Testbed. It is likely that either the E43 element of the Testbed or one of the new formulations installed via the Generic Element Processor will be used for any 2-D analyses which are required. Investigation will also be made into the use of classical lamination theory results for generating strains and curvatures, which can then be converted to displacements and applied to FE models.

The target problem is the CSM Focus Problem, a composite plate with blade stiffeners, with at least one stiffener interrupted by a hole. Of primary interests in the composite mechanics are the discontinuities, the regions near the hole and the juncture of the stiffeners and the plate. Detailed 3-D analyses will be conducted in these regions in order to understand the mechanism of load transfer and hence failure.

In addition to rules for 3-D analysis of laminates, guidelines will be developed which can be used for the corresponding global/local analyses. These will include guidelines for estimating the proper location of the global/local interface, and for assessing the validity of solutions once that decision has been made. An attempt will be made to develop semi-automatic techniques for location of the interface and also for examination of the solution to issue a report to the user which would point out any regions where the solution might be questionable.

REFERENCES

1. R. M. Barker, F. T. Lin, and J. R. Dana, "Three-dimensional Finite-element Analysis of Laminated Composites," *Computers and Structures*, Vol 2, 1972, pp 1013-1029.
2. O. H. Griffin, Jr., M. P. Kamat, and C. T. Herakovich, "Three-dimensional Inelastic Finite Element Analysis of Laminated Composites," *Journal of Composite Materials*, Vol 15, Nov 1981, pp 543-560.
3. S. W. Burns, C. T. Herakovich, and J. G. Williams, "Compressive Failure of Notched Angle-Ply Composite Laminates: Three Dimensional Finite Element Analysis and Experiment," VPI&SU Report VPI-E-85-22 (CCMS-85-11), Nov 1985.
4. C. T. Herakovich, D. Post, M. B. Buczek, and R. Czarnek, "Free Edge Strain Concentrations in Real Composite Laminates: Experimental-Theoretical Correlation," VPI&SU Report VPI-E-84-24 (CCMS-84-24), Jul 1984.
5. S. R. Mathison, C. T. Herakovich, M. J. Pindera, and M. J. Shuart, "Nonlinear Analysis of the Response and Failure of Compression-Loaded Angle-Ply Laminates with a Hole," VPI&SU Report VPI-E-87-14 (CCMS 87-12), Jul 1987.
6. C. T. Sun and P. Mehrotra, "Global-Local Structural Analysis Using a Multiprocessor Computer," in the proceedings of the NASA Langley CSM Review, August 25, 1986, W. J. Stroud, Editor.
7. E. L. Stanton, L. M. Crain, and T. F. Neu, "A Parametric Cubic Modelling System for General Solids of Composite Material," *International Journal for Numerical Methods in Engineering*, Vol 11, 1977, pp 653-670.
8. S. W. Burns, S. R. Mathison, and C. T. Herakovich, "ANISAP--A Three Dimensional Finite Element Program for Laminated Composites Subjected to Mechanical Loading," VPI&SU Report VPI-E-86-12 (CCMS 86-03), Jul 1986.
9. R. J. Melosh, "Development of the Stiffness Method to Define Bounds on Elastic Behavior of Structures," Ph.D. Dissertation, University of Washington, 1962.
10. F. T. Lin, "The Finite Element Analysis of Laminated Composites," Ph.D. Dissertation, Virginia Polytechnic Institute & State University, 1972.
11. C. S. Desai and J. F. Abel, Introduction to the Finite Element Method, Van Nostrand Reinhold, New York, 1972.
12. R. H. MacNeal and R. L. Harder, "A Proposed Standard Set of Problems to Test Finite Element Accuracy," *Finite Elements in Analysis and Design*, Vol 1, 1985, pp 3-20.
13. A. K. Noor, "Global-Local Methodologies and Their Application to Nonlinear Analysis," *Finite Elements in Analysis and Design*, Vol 2, 1986, pp 333-346.
14. C. T. Herakovich, "Influence of Layer Thickness on the Strength of Angle-Ply Laminates," *Journal of Composite Materials*, Vol 16, May 1982, pp 216-227.

N89
29798
UNCLAS

N89-29798

CONTROL OF THE ERRORS OF DISCRETIZATION AND IDEALIZATION IN FINITE ELEMENT ANALYSIS†

Barna A. Szabó
Center for Computational Mechanics
Washington University, St. Louis, MO 63130

Introduction.

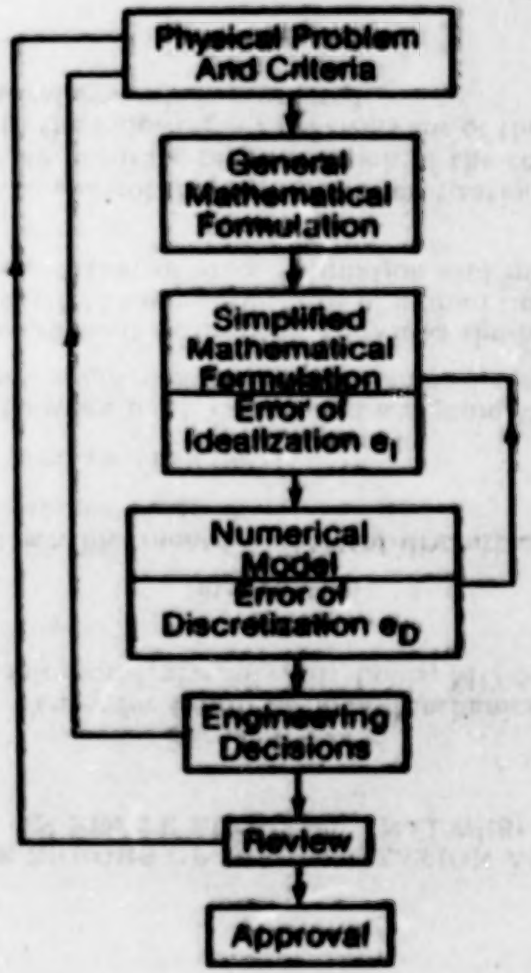
Our understanding of the basic principles which control errors of discretization in finite element analysis has increased very substantially since 1980. The main milestones were:

1. Development of the theoretical basis of p-extensions (1981);
2. Understanding of the proper interplay between mesh design and assignment of polynomial degree to elements. Practical realization of exponential convergence rates, independently of the smoothness of the exact solution (1984);
3. Industrial experience with the new finite element technology known as the p- or hp-version of the finite element method: General Dynamics reported thirty- to forty-fold savings in terms of human time and large savings in computer time (1986). Lockheed reported favorably on their evaluation of error estimation and quality control capabilities of the p-version in industrial setting (1987). See [1,2].

The gains in our understanding of how to control the errors of discretization represent only half of the control necessary to ensure that a numerical model is in fact an accurate representation of the corresponding physical system. Control of the *errors of idealization* is equally important. In the following a brief overview of the main ideas of how to ensure the quality and reliability of mathematical models of structural systems is presented.

† *This work is supported by NASA Grant Number NAG-1-693.*

THE ERRORS OF IDEALIZATION AND DISCRETIZATION



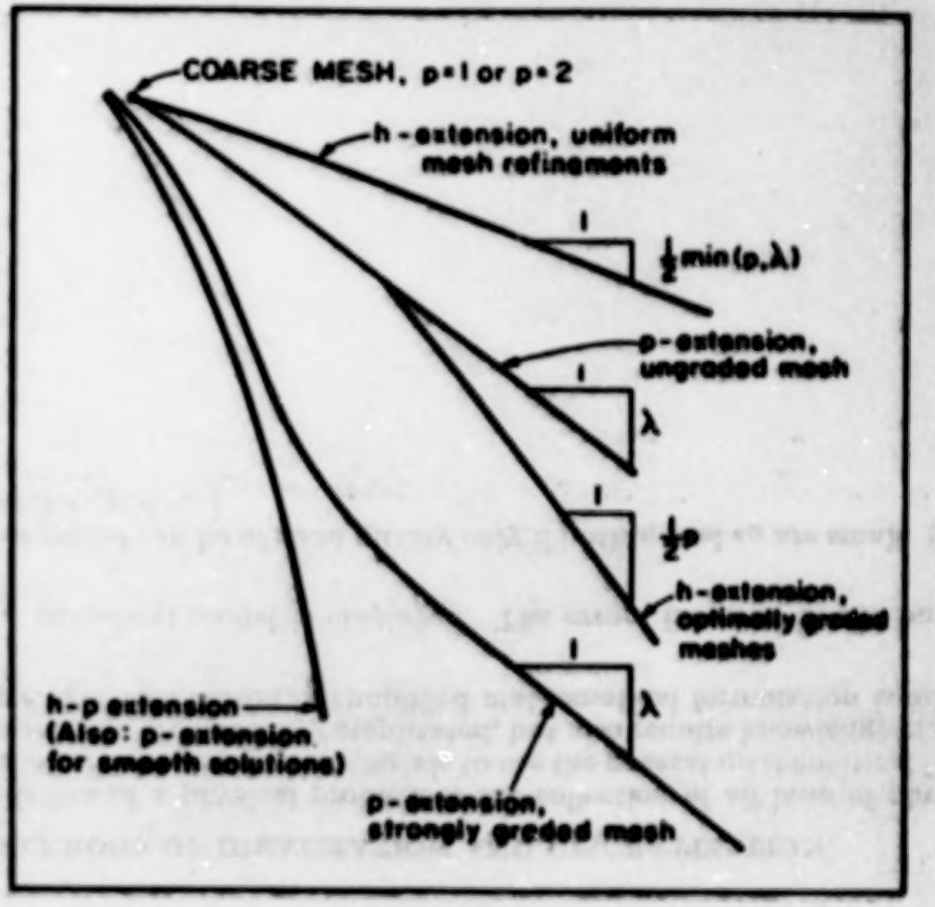
THE ERRORS OF IDEALIZATION AND DISCRETIZATION.

The general mathematical formulation of a physical problem is the collection of all laws of physics which apply to that problem. Unless the physical problem is very simple, it is not feasible to use the general mathematical formulation in engineering decision making processes. Such formulations are not only complicated, but also require knowledge of physical properties which are difficult and expensive to measure. For this reason, a simplified mathematical formulation is adopted, thereby incurring an error of idealization e_I.

To solve the physical problem, a numerical model is employed. The errors incurred in the numerical model are called errors of discretization e_D.

Engineering decisions based on the model can be of good quality only if both e_I and e_D are small. (We assume that roundoff errors are negligible in comparison with e_I and e_D.)

RATES OF CONVERGENCE OF h-, p-, AND hp-EXTENSIONS



RATES OF CONVERGENCE OF h-, p-, AND hp-EXTENSIONS

For a given formulation, such as the displacement formulation, finite element solutions are characterised by the finite element mesh, the polynomial degree of elements, and the mapping functions. We can create a sequence of finite element solutions by systematically refining the mesh or increasing the polynomial degree of elements, or both. The process of creating sequences of finite element solutions is called *extension*. If the extension is by mesh refinement, it is called *h-extension*; if the extension is by increase of the polynomial degree then it is called *p-extension*; if the extension is by combination of mesh refinement and increase of the polynomial degree then it is called *hp-extension*.

Errors of discretization are controlled by extensions. Properly performed extensions permit us to assess and control the quality of finite element solutions. Specifically, the following steps are recommended:

(1) Estimate the relative error in the energy norm. This is equivalent to estimating the *root-mean-square error of stresses*. Therefore estimates of error in energy norm are indicators of the overall quality of the solution. Usually, however, we are interested in quantities which are associated with points and lines in two dimensions; points, lines and surfaces in three dimensions. The error in these quantities can be large even when the error in energy norm is small. Thus additional quality control measures are necessary.

(2) Check equilibrium. It should be possible to draw free body diagrams for the entire structure or any part of the structure to check that equilibrium is satisfied to within a tolerance level which is a small fraction of the total applied load.

(3) Check whether the action-reaction principle is satisfied to within acceptable levels of tolerance.

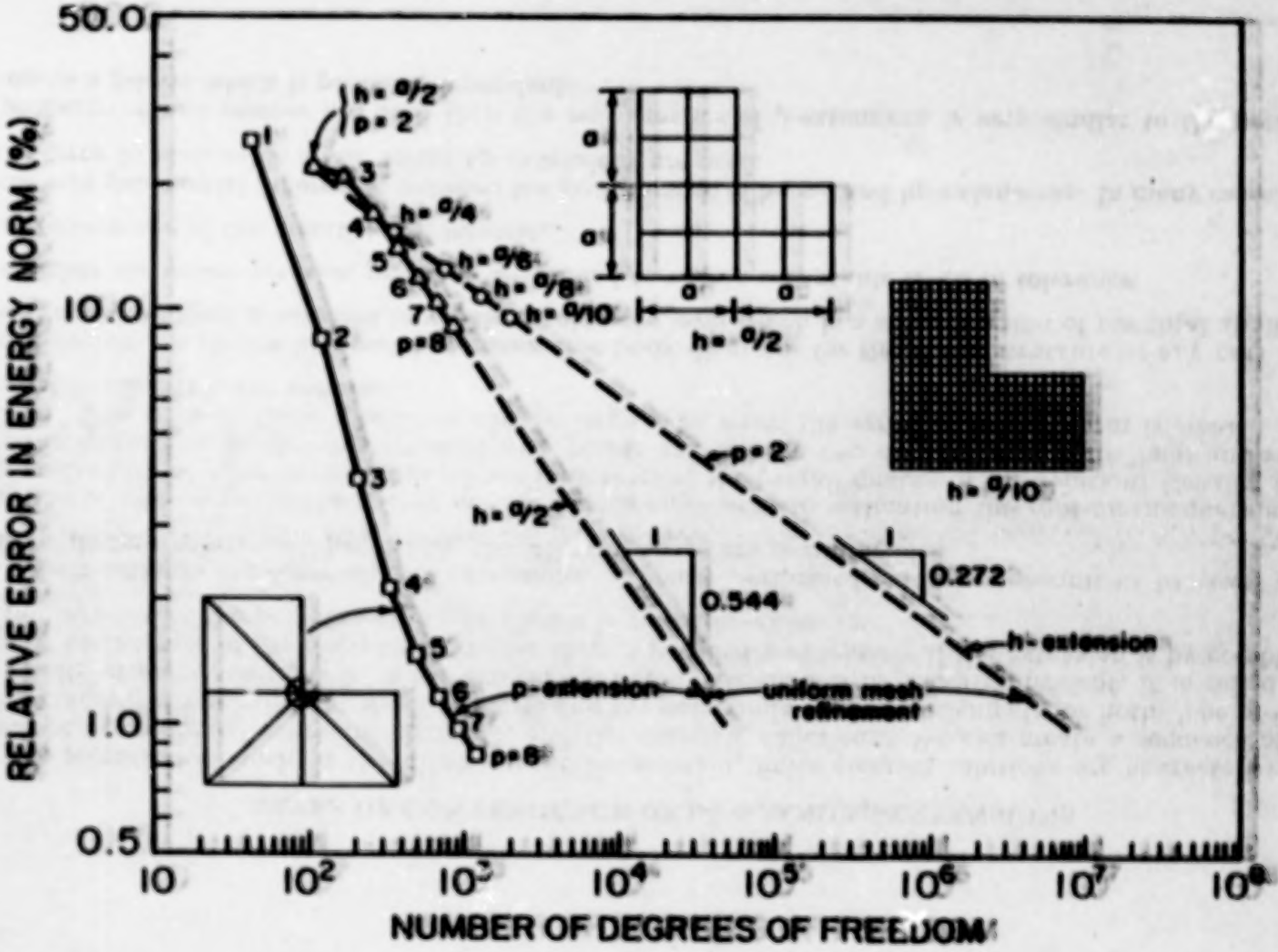
(4) Observe convergence of the quantities of interest.

There are very substantial differences between the performance of h-, p-, and hp-extensions. In many cases it is not feasible to reduce the error to reasonable levels unless hp-extensions are used.

When properly refined meshes are used then the performance of p-extensions is very similar to the performance of hp-extensions up to a p-level which is problem dependent.

EXAMPLE: THE L-SHAPED DOMAIN.

CONVERGENCE OF h- AND p-EXTENSIONS.



EXAMPLE: THE L-SHAPED DOMAIN. CONVERGENCE OF h- AND p-EXTENSIONS.

The model problem of the L-shaped domain is typical of many problems where corner singularities occur. This problem was constructed so that the exact solution is known. Specifically, the imposed loading was by tractions computed from the following stress components:

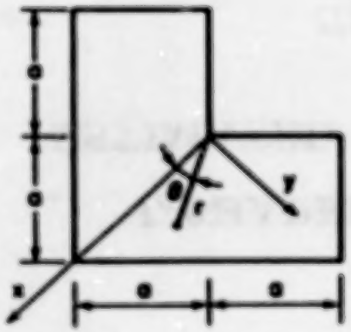
$$\sigma_x = \lambda r^{\lambda-1}[(2 - Q(\lambda+1))\cos(\lambda-1)\theta - (\lambda-1)\cos(\lambda-3)\theta] \quad (1)$$

$$\sigma_y = \lambda r^{\lambda-1}[(2 + Q(\lambda+1))\cos(\lambda-1)\theta + (\lambda-1)\cos(\lambda-3)\theta] \quad (2)$$

$$\tau_{xy} = \lambda r^{\lambda-1}[(\lambda-1)\sin(\lambda-3)\theta + Q(\lambda+1)\sin(\lambda-1)\theta]. \quad (3)$$

where r, θ are polar coordinates centered on the reentrant corner, as shown in the figure below; $\lambda = 0.5444483737$ and $Q = 0.5430755579$. In the finite element solutions plane strain conditions, and Poisson's ratio of 0.3 were assumed. For additional details see [3]. Note that to achieve 1% relative error in energy norm with h-extension, uniform mesh refinement and $p = 2$, 10^7 degrees of freedom are needed. The same accuracy can be achieved with p-extension on a radical mesh, with 10^3 degrees of freedom.

Assuming that the solution is by Gaussian elimination with optimal ordering of the operations, the work is proportional to $N^{3/2}$ where N is the number of degrees of freedom. Thus, the work ratio is of the order of 10^6, which is about 1 minute to 2 years.



L-SHAPED DOMAIN. p-EXTENSION ON RADICAL MESH.

ESTIMATED AND TRUE RELATIVE ERRORS IN ENERGY NORM.

p	N	$\frac{U(\vec{u}_{FE})E}{\sigma_\infty^2 a^2 t_x}$	$\frac{U^*(\vec{u}_{EX})E}{\sigma_\infty^2 a^2 t_x}$	Est.'d 2β	True 2β	Est.'d $(e_r)_E$ (%)	True $(e_r)_E$ (%)
1	41	3.8860880	–	–	–	25.42	25.42
2	119	4.1248326	–	2.07	2.07	8.45	8.46
3	209	4.1481150	4.1599969	2.74	2.72	3.91	3.93
4	335	4.1526504	4.1543418	2.65	2.59	2.09	2.14
5	497	4.1536354	4.1540572	1.99	1.87	1.42	1.48
6	695	4.1539746	4.1542413	1.51	1.35	1.09	1.17
7	929	4.1541390	4.1543718	1.43	1.21	0.89	0.99
8	1199	4.1542378	4.1544700	1.42	1.12	0.75	0.86
∞	∞	4.1545442	4.1545442	1.09	1.09	0	0

L-SHAPED DOMAIN. p-EXTENSION ON RADICAL MESH. ESTIMATED AND TRUE RELATIVE ERRORS IN ENERGY NORM.

The relative error in energy norm is estimated from three consecutive solutions by solving for the exact strain energy $U_{EX} \overset{\text{def}}{=} U(\bar{u}_{EX})$ in the theoretical estimate:

$$|U_{EX} - U_p| \cong \frac{k^2}{N_p^{2\beta}}$$

where U_p is the strain energy computed from the finite element solution corresponding to degree p, N_p is the number of degrees of freedom, k and β are constants, 2β is called the asymptotic rate of convergence of the strain energy.

The estimated relative error in energy norm defined by:

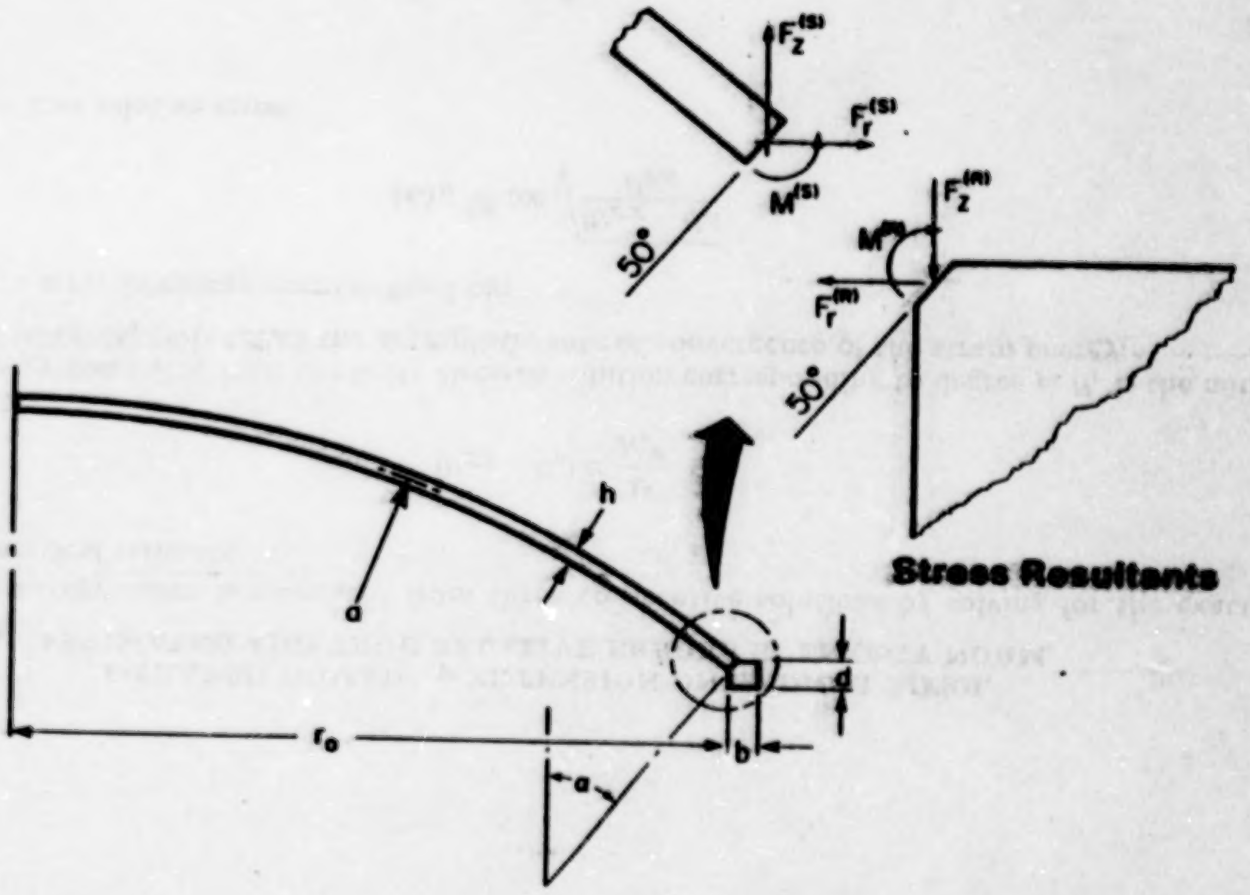
$$(e_r)_E \overset{\text{def}}{=} 100\sqrt{\frac{|U_{EX} - U_p|}{U_{EX}}}$$

is reasonably close to the true relative error.

GIRKMANN'S PROBLEM: SPHERICAL SHELL WITH EDGE RING.

$a = 919.2$ in; $h = 2.36$ in; $\alpha = 40°$; $b = 23.64$ in; $d = 19.68$ in

(Not to scale.)



GIRKMANN'S PROBLEM: SPHERICAL SHELL WITH EDGE RING

To demonstrate control of the error of idealization, let us consider the problem of a spherical shell with an edge ring posed by Girkmann in 1956 [4] and quoted by Timoshenko and Woinowsky-Krieger in 1959 [5]. The shell is loaded by gravity which was modeled by Girkmann as uniformly distributed vertical load acting on the middle surface. The goal of computation is to determine the reactions F_r, and M. The following results were obtained by Girkmann:

$$F_r = -8.95 \text{ lbf/in.}, \qquad M = -24.84 \text{ lbf in./in.}$$

He idealized the edge ring as a circular beam, estimated the radial and torsional stiffnesses of this ring and computed the reactions by equating the displacements of the ring and the shell.

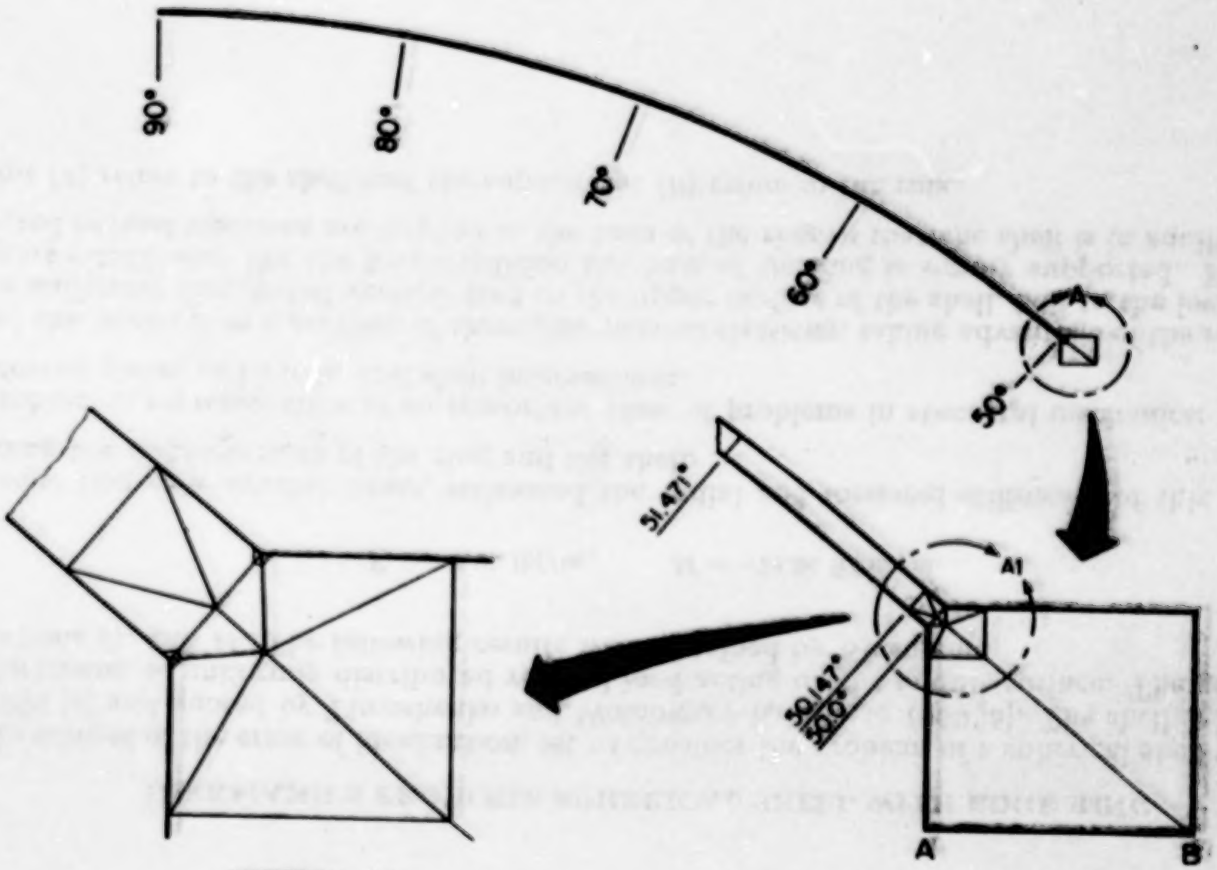
This model problem is representative of an important class of problems in stuctural mechanics: problems which involve stiffened and reinforced plates and shells, and shell intersections.

We have solved this problem as a problem of three dimensional elasticity, taking advantage of the rotational symmetry. We applied half of the uniformly distributed vertical load to the upper surface of the shell, half to the lower surface. We assumed two kinds of support conditions: For the first condition the base of the ring is simply supported. For the second condition uniformly distributed normal tractions are applied to the base of the ring so that the shell is in equilibrium.

The superscript (S) refers to the shell and the superscript (R) refers to the ring.

GIRKMANN'S PROBLEM: SPHERICAL SHELL WITH EDGE RING.

22-element mesh.



Mesh M22

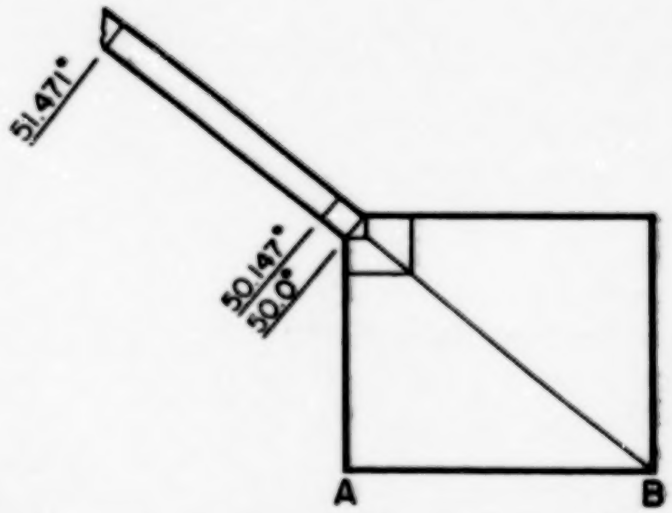
GIRKMANN'S PROBLEM: SPHERICAL SHELL WITH EDGE RING.
22-element mesh.

This figure, drawn to scale, shows the true aspect ratios in Girkmann's problem. The radius to thickness ratio is 389.5. Essentially, the spherical shell is connected to a solid body.

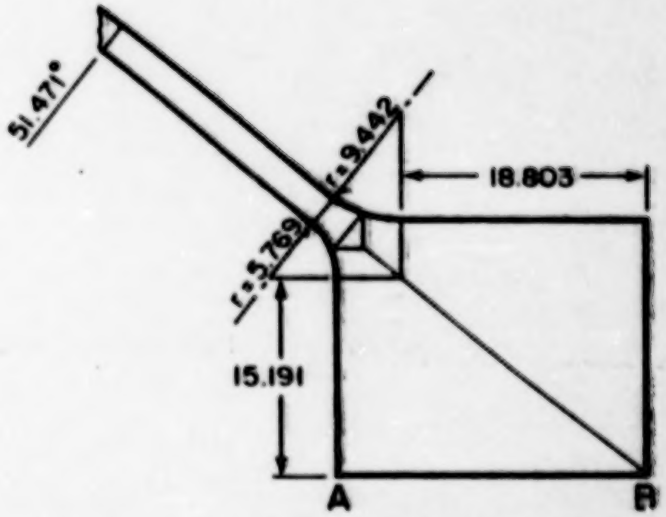
This figure also shows one of the mesh designs, called Mesh M22. Note the similarity in grading with the L-shaped domain.

GIRKMANN'S PROBLEM: SPHERICAL SHELL WITH EDGE RING.

Two 11-element meshes. Detail of the shell-ring junction.



Mesh M11



Mesh M11F

GIRKMANN'S PROBLEM: SPHERICAL SHELL WITH EDGE RING
Two 11-element meshes. Detail of the shell-ring junction.

This figure shows finite element meshes for the shell-ring junction. Both meshes consist of 11 finite elements. In the case of mesh M11, the junction has two sharp reentrant corners. In the case of mesh M11F, the corners have been rounded by fillets as shown. The exact solution is much smoother in the case of mesh M11F than in the case of mesh M11.

GIRKMANN'S PROBLEM: SPHERICAL SHELL WITH EDGE RING.

Estimated relative errors in energy norm $(e_r)_E$ (percent).

p	N	Mesh M11	Mesh M11F	N	Mesh M22
1	40	54.56	54.95	56	55.36
2	101	18.12	18.11	156	18.09
3	164	6.71	6.51	274	6.52
4	249	4.54	4.36	436	4.08
5	356	2.42	2.07	642	1.98
6	485	1.17	0.57	892	0.54
7	636	0.80	0.16	1186	0.11
8	809	0.57	0.05	1524	0.03

GIRKMANN'S PROBLEM: SPHERICAL SHELL WITH EDGE RING
Estimated relative errors in energy norm $(e_r)_E$.

The estimated relative errors in the energy norm are less than one percent for $p \geq 7$ for all three meshes, as shown in the table. If we were to judge the solutions only on the basis of the error in the energy norm, then we would have no reason to reject either sequence of solutions.

STRESS RESULTANTS AT THE SHELL-RING INTERFACE

Mesh M11. Simple support.

p	$F_r^{(S)}$ (lbf/in)	$F_r^{(R)}$ (lbf/in)	$F_z^{(S)}$ (lbf/in)	$F_z^{(R)}$ (lbf/in)	$M^{(S)}$ (in lbf/in)	$M^{(R)}$ (in lbf/in)
2	−84.6	−86.6	100.1	43.0	−234.5	−75.2
3	−92.6	−84.8	95.3	49.5	−235.7	−112.8
4	−92.9	−91.4	95.3	55.8	−234.2	−123.8
5	−93.1	−90.0	95.3	63.5	−230.0	−157.9
6	−93.1	−90.7	95.3	69.6	−229.2	−172.2
7	−93.1	−91.2	95.3	74.3	−229.0	−182.4
8	−93.1	−91.4	95.3	77.5	−229.0	−189.9
∞	(?)	(?)	95.3	95.3	(?)	(?)

STRESS RESULTANTS AT THE SHELL-RING INTERFACE
Mesh M11. Simple support.

The superscripts (S) and (R) refer to whether the resultants were computed for the shell or the ring, respectively.

The value of F_z is known from equilibrium. It is 95.3 lbf/in. Obviously F_z, computed from the finite element solutions, must converge to this value. The values of F_r and M are not known. Therefore the corresponding limit values, indicated by question marks, must be inferred from the sequence of values computed from the finite element solutions.

It is always possible to compute at least some functionals from the finite element solutions, the exact values of which are known, or can be computed from the input data. In this example F_z is such a functional. Such functionals are useful indicators of the quality of similar functionals computed from the finite element solutions.

This table shows that the action-reaction principle is not satisfied at p=8 for the vertical forces ($F_z^{(S)} \neq F_z^{(R)}$) and the moments ($M^{(S)} \neq M^{(R)}$). Nevertheless, very slow convergence is evident. The radial force F_R very nearly satisfies the action-reaction principle. Interestingly, $F_z^{(S)}$ is very accurate.

STRESS RESULTANTS AT THE SHELL-RING INTERFACE

Mesh M11F. Simple support.

p	$F_r^{(S)}$ (lbf/in)	$F_r^{(R)}$ (lbf/in)	$F_z^{(S)}$ (lbf/in)	$F_z^{(R)}$ (lbf/in)	$M^{(S)}$ (in lbf/in)	$M^{(R)}$ (in lbf/in)
2	−114.0	−95.2	47.0	90.2	−253.9	−143.9
3	−91.8	−90.6	97.6	97.3	−249.6	−254.8
4	−91.3	−93.2	96.4	96.0	−249.3	−258.8
5	−92.4	−93.1	95.1	94.7	−244.7	−244.1
6	−92.3	−92.1	95.3	95.6	−243.9	−245.4
7	−92.3	−92.4	95.3	95.1	−243.9	−243.2
8	−92.3	−92.3	95.3	95.3	−243.9	−244.2
∞	(?)	(?)	95.3	95.3	(?)	(?)

STRESS RESULTANTS AT THE SHELL-RING INTERFACE
Mesh M11F. Simple support.

This table shows that introducing fillets in the numerical model has a very large effect on the quality of the solution not only from the point of view of convergence in the energy norm but also from the point of view of satisfying the action-reaction principle. The question arises: Which is the better idealization, the stiffened shell with the rounded fillets or the sharp corners? Clearly, use of the rounded fillets simplifies the analysis considerably.

Note that F_r, F_s and M satisfy the action-reaction principle to three significant digits for $p = 8$. The moments deviate from their average by only 0.06%.

STRESS RESULTANTS AT THE SHELL-RING INTERFACE

Mesh M22. Simple support.

p	$F_r^{(S)}$ (lbf/in)	$F_r^{(R)}$ (lbf/in)	$F_x^{(S)}$ (lbf/in)	$F_x^{(R)}$ (lbf/in)	$M^{(S)}$ (in lbf/in)	$M^{(R)}$ (in lbf/in)
2	−73.3	−94.8	129.3	75.1	−253.2	−214.1
3	−92.9	−90.4	86.8	88.3	−240.6	−218.5
4	−95.5	−91.4	86.6	90.4	−227.9	−221.8
5	−95.5	−92.2	92.6	90.2	−230.9	−221.1
6	−93.7	−92.5	95.9	91.9	−230.2	−222.1
7	−92.8	−92.6	96.0	92.5	−228.8	−223.2
8	−92.9	−92.7	95.4	92.9	−228.5	−224.1
∞	(?)	(?)	95.3	95.3	(?)	(?)

STRESS RESULTANTS AT THE SHELL-RING INTERFACE
Mesh M22. Simple support.

This table shows that isolation of the reentrant corners by rings of small elements significantly improves the quality of the solution judged from the points of view of satisfaction of equilibrium and the action-reaction principle. Nevertheless, addition of the rings is not as effective as introduction of the fillets, even though the number of degress of freedom increased from 809 to 1524 at p=8.

In particular, if we assume once again that the solution is performed by Gaussian elimination, with optimal ordering of the operations, the work ratio in the solution phase is $(1524/809)^{1.5} = 2.6$. The work ratio in the stiffness generation phase is $22/11 = 2$. The work in the input phase, the human time, is generally the costliest operation but also the most difficult one to quantify. Nevertheless, this also favors mesh M11F to mesh M22.

STRESS RESULTANTS AT THE SHELL-RING INTERFACE

Mesh M22. Equilibrium loading.

p	$F_r^{(S)}$ (lbf/in)	$F_r^{(R)}$ (lbf/in)	$F_z^{(S)}$ (lbf/in)	$F_z^{(R)}$ (lbf/in)	$M^{(S)}$ (in lbf/in)	$M^{(R)}$ (in lbf/in)
5	−105.4	−103.7	95.1	93.6	−7.89	−7.02
6	−105.7	−104.0	95.7	93.9	−7.88	−7.14
7	−105.3	−104.2	95.4	94.2	−7.79	−7.24
8	−105.1	−104.4	95.2	94.3	−7.76	−7.32
∞	(?)	(?)	95.3	95.3	(?)	(?)

STRESS RESULTANTS AT THE SHELL-RING INTERFACE

Mesh M22. Equilibrium loading.

When the shell is held in equilibrium by uniformly-distributed normal tractions applied at the base of the ring, the radial force and the moment change very significantly as compared with the simply supported condition. The question arises: Which idealization of the support is more nearly representative of the support of the real shell? Usually, the support has elastic properties. The sensitivity of the stress resultants to support conditions indicates that the elastic properties of the support must be taken into consideration.

GIRKMANN'S PROBLEM

Summary of results.

	Force F_r (lbf)	Moment M (lbf in/in)
Girkmann/Timoshenko :	−8.95	−24.84
Finite element analysis :		
(a) Simple support	-92.8 ± 0.1	-226.3 ± 2.2
(b) Equilibrium loading	-104.8 ± 0.4	-7.54 ± 0.22

GIRKMANN'S PROBLEM
Summary of Results.

This example clearly shows that certain idealizations can lead to very bad results even in the hands of internationally recognized experts. The difficulty is caused by the fact that the stiffness of the ring cannot be computed with sufficient accuracy by the simple methods of idealization based on standard techniques of structural engineering.

We claim only that the results of the finite element analyses are accurate with respect to solutions based on the three dimensional theory of elasticity. The tolerance levels indicated are estimates based on the observed discontinuity of stress resultants.

We have not considered the effects of deformation on equilibrium or the possibility of yielding. Proper control of the errors of idealization would require us to investigate whether all of the assumptions incorporated in the linear theory of elasticity hold in this particular case. Since we modeled the shell-ring junction with sharp corners, we know that at least some local yielding may occur. In view of the fact that the shell is very thin, we expect geometric nonlinearities to have a significant effect on the computed data.

SUMMARY AND CONCLUSIONS.

(1) Errors of discretization are estimated and controlled by extensions.

(2) Errors of idealization are also estimated and controlled by extensions. Theoretical representations of physical systems constitute a natural hierarchic order. We seek the lowest level in this order which accounts for all essential details.

(3) The use of sharp reentrant corners should be avoided. It is generally less expensive to account for fillets than to omit them.

(4) Smallness of error in energy norm does not quarantee that the solution is good. We must also investigate equilibrium, satisfaction of the action-reaction principle and convergence of the quantities of interest.

SUMMARY AND CONCLUSIONS.

(1) Errors of discretization are estimated and controlled by extensions: A *hierarchic sequence of discretizations* is created and the corresponding finite element solutions are computed. The error in energy norm can be estimated accurately. The error in other quantities is estimated by observing convergence to their limiting values. P-extensions and properly designed meshes provide the most efficient and most reliable means for error estimation and quality control procedures. For surveys of the key theoretical results see [6,7]. For additional engineering data see [3,8,9, 10].

(2) Errors of idealization are also estimated and controlled by extensions: The various theories constitute a natural hierarchic order. In structural mechanics, the lowest levels of the hierarchy are the engineering theories of strength of materials. The highest levels are fully three-dimensional representations which account for all geometric details, material and geometric nonlinearities; the mechanical properties of support conditions, etc. In general, we first solve a problem by the simplest available means, much the same way as Girkmann solved the stiffened shell problem. But then we must continue to higher levels of idealization until we can ascertain that no essential details have been overlooked. We have shown that the Girkmann/Timoshenko model is in substantial error with respect to a fully three-dimensional model based on the theory of elasticity. Our solution cannot be accepted as an accurate representation of the structural response of this shell until we estimate the effect of geometric nonlinearities and check whether the stresses exceed the elastic limit. We have seen that the mechanical properties of the support have a large effect on the stress resultants at the shell-ring intersection.

(3) Sharp corners rather than fillets are often used in finite element models because analysts believe that minor topological details do not affect the structural response significantly. While this is usually true, the performance of h- and p-extensions is significantly affected by singularities. We have seen that it is better to avoid the use of sharp reentrant corners than to compensate for the consequent discretization error by mesh refinement. We note that real structures usually do not have sharp corners.

(4) Smallness of error in energy norm does not guarantee that the quality of the finite element solution is good when the purpose of computation is other than determination of the strain energy. In checking the quality of finite element solutions it is essential to investigate equilibrium, satisfaction of the action-reaction principle, and convergence of the quantities of interest.

REFERENCES.

[1] Barnhart, M. A. and Eisenmann, J. R., "Analysis of a Stiffened Plate Detail Using p-Version and h-Version Finite Element Techniques", Paper presented at the First World Congress on Computational Mechanics, Sept. 22-26, 1986, The University of Texas at Austin.

[2] Brussat, T. R. and Malone, R. L., "Stress Analysis of Lugs Using Probe" Report LR 31229, Lockheed California Company (1987)

[3] Szabó, B. A., "Mesh Design for the p-Version of the Finite Element Method", Computer Methods in Applied Mechanics and Engineering, Vol. 55, pp. 181-197 (1986).

[4] Girkmann, K., *Flächentragwerke*, Fourth Edition, Springer-Verlag, Wien, pp. 442-447 (1956).

[5] Timoshenko, S. and Woinowsky-Krieger, S., *Theory of Plates and Shells*, Second Edition, McGraw-Hill, pp. 555-558 (1959).

[6] Babuška, I., "The p- and hp-Versions of The Finite Element Method. The State of the Art", *Finite Elements: Theory and Application*, edited by D. L. Dwoyer, M. Y. Hussaini and R. G. Voigt, Springer (1988).

[7] Suri, M., "The p-Version of the Finite Element Method for Elliptic Problems" *Advances in Computer Methods for Partial Differential Equations - VI*, R. Vichnevetsky and R. S. Stepleman, Editors, Int.'l Association for Mathematics and Computers in Simulation, pp. 85-91 (1987).

[8] Szabó, B., "Estimation and Control of Error Based on P-Convergence" in: I. Babuška, J. Gago, E. R. de A. Oliveira and O. C. Zienkiewicz, editors, *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, John Wiley & Sons Ltd., pp. 61-78, (1986).

[9] Rank, E. and Babuška, I., "An Expert System for the Optimal Mesh Design in the hp-Version of the Finite Element Method", *Int. J. for num. Meth. Engng.*, Vol. 24, pp. 2087-2106 (1987).

[10] Szabó, B. A. and Sahrmann, G. J., "Hierarchic Plate and Shell Models Based on p-Extension", Report WU/CCM-87/5, Center for Computational Mechanics, Washington University, St. Louis, MO 63130 (1987). To appear in: *Int. J. for num. Meth. Engng.*

NASA

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CP-10012, Part 2		

4. Title and Subtitle	5. Report Date
NASA Workshop on Computational Structural Mechanics - 1987	February 1989
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Nancy P. Sykes, Editor	
	10. Work Unit No. 505-63-01-10

9. Performing Organization Name and Address	11. Contract or Grant No.
NASA Langley Research Center Hampton, VA 23665-5225	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Conference Publication
National Aeronautics and Space Administration Washington, DC 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

Nancy P. Sykes: Analytical Services and Materials, Inc., Hampton, Virginia.

16. Abstract

This conference publication contains the proceedings of the Workshop on Computational Structural Mechanics held at NASA Langley Research Center, November 18-20, 1987. The workshop was sponsored jointly by NASA Langley Research Center and NASA Lewis Research Center.

The workshop was organized into the following three sessions:

(1) Concurrent Processing Methods and Applications
(2) Advanced Methods & Testbed/Simulator Development
(3) Computational Dynamics

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Parallel Processing Computational Dynamics Testbed Computational Structural Mechanics	Unclassified - Unlimited Subject Category 39

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	382	A17

NASA FORM 1626 OCT 86 For sale by the National Technical Information Service, Springfield, Virginia 22161-2171

END

DATE

FILMED

DEC 1 1989